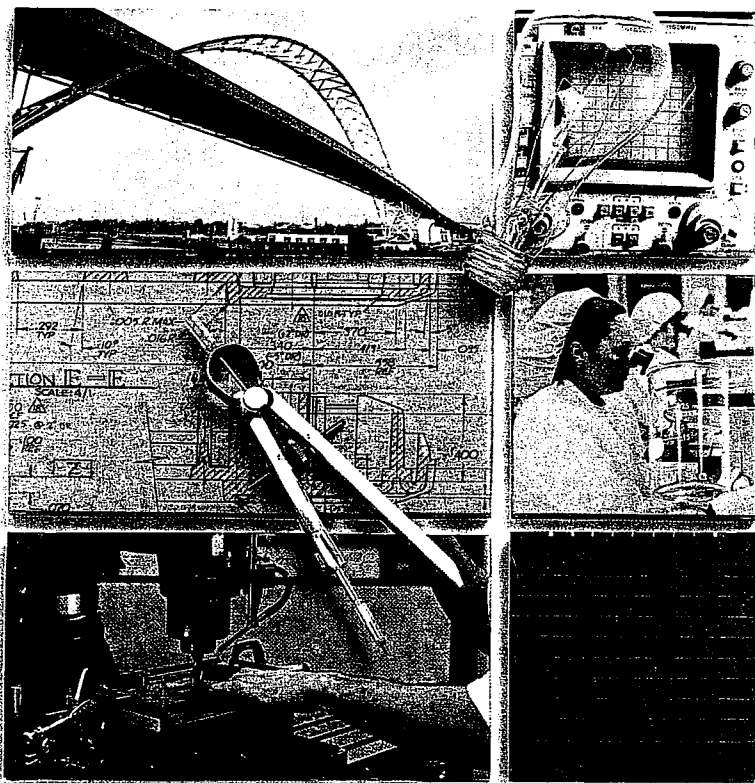


横河・ヒューレットパカード株式会社

HP-11C

操作ハンドブック

+ プログラム作成の手引き



御 注 意

当社はこの本に記載したキー操作順序やプログラム内容に関しましては、どんな場合にでも必ず正しい答を求められるとは保証いたしておりません。特殊な場合には別のキー操作や別のプログラムが必要になることがあります。この本のキー操作順序やプログラムはこのような計算のときにはこのようにするのだという一例にすぎません。従って取引上の計算などにはご使用の方が広範囲にテストした上で利用するかどうか決めてくださるようお願いいたします。またそのためにプログラム等を改良する必要がありましたときは、ご使用の方の責任と費用で改良してくださるようお願いいたします。ご使用の方がこの通りに操作したために発生した逸失利益・金銭上の損害や第三者からの請求を受けることがありましても、当社はその責任を負いませんので御了承くださるようお願いいたします。

 横河・ヒューレットパカード株式会社

HP-11C

操作ハンドブック

+プログラム作成の手引き

1987年1月

部品番号
00011-90008
Edition 1

はじめに

どうもありがとうございました。あなたのお選びになった HP-11C は不揮発性メモリー付きなので、便利で使いやすく高性能な計算機です。この本はまだ計算機におなじみでない方のために性能や使い方を簡単に説明したものです。

基礎知識、プログラム作成、各種のプログラム この本は三つに大別してあります。第1部と第2部はキー操作の方法や、プログラム作成法なので、いままでに当社の他のプログラム電卓をお使いになったことがあればかなりご存じのことが多いと思います。第3部には10種のプログラムと、プログラム作成の技術的ヒントをまとめてあります。かなりお知りになっている方も9ページの HP-11C を使ってみましょうからお読みになることをお勧めします。

プログラム作成の要点 第2部（5章から9章まで）の各章の初めにその章で説明することの要点がまとめてあります。この要点は当社の他のプログラム電卓をお使いになった方も必ず読んでいただきたいと思います。当社の電卓は HP-11C が初めてという方はこの要点だけでなくそれ以降の説明を読んで、例題などを実際に操作してみてください。

その他のプログラム 当社では従来の機種用のプログラムを HP-11C 用に組替えて、HP-11C アプリケーション・ブック（製品番号 00011-90040）として 81 年 12 月に発売するよう準備中です。数学・統計・電気・機械など 150 点以上を収めたプログラム集になる予定です。

目 次

はじめに	2
HP-11C のキーと不揮発性メモリー	8
HP-11C を使ってみましょう	9
手操作	10
プログラム計算	12
第 1 部 HP-11C の基礎知識	15
第 1 章 まず始めましょう	16
スイッチの入・切	16
表示	16
小数点と桁区切	16
状態表示	16
負数	17
表示の取消し	17
走行表示	18
オーバーフローとアンダーフロー	18
エラー表示	19
低電圧表示	19
メモリー	19
不揮発性メモリー	19
メモリーのリセット	20
キー操作	20
キーの第一機能と第二機能	20
前置キーの取消し	21
単項演算キー	21
二項演算キー	22
第 2 章 自動メモリー・スタック、ラスト x、数値のストア	26
自動メモリー・スタックとスタック操作	26
スタック操作	27
関数計算とスタック	29
二項演算	29
長い計算	31
ラスト x	32
定数計算	34

4 目 次

記憶レジスタの操作	37
数値の記憶 (ストア)	37
記憶させた数値の呼出し (リコール)	38
ストアとリコールの練習	38
記憶レジスタのクリア	38
記憶レジスタの直接四則演算	39
記憶レジスタの直接四則演算の練習	40
練習問題	40
第 3 章 関数	42
π (円周率)	42
数値変更用	42
単項演算関数	43
普通の関数	43
三角関数	45
時間と角度の変換	46
度とラジアン相互変換	47
対数	48
双曲線関数	48
二項演算関数	49
累乗	49
パーセント	49
極座標と直交座標の変換	51
順列と組合せ	52
統計用関数	54
乱数発生	54
統計用データの集計	55
集計データの訂正	58
平均	60
標準偏差	61
直線のあてはめ	63
y の推定と相関係数	64
第 4 章 表示指定	67
表示形式の指定	67
固定小数点表示	67
浮動小数点表示	68
工学浮動小数点表示	70
大きな数値のキーイン	71
10 桁目の数字	72

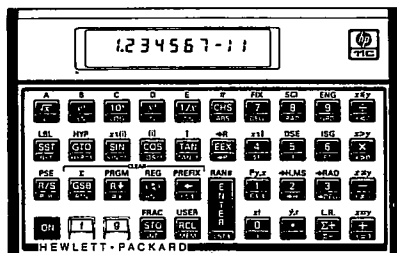
第 2 部	HP-11C のプログラム	73
第 5 章	プログラム作成の基礎知識	74
	プログラムとは	74
	なぜプログラムを書く必要があるのでしょうか	74
	プログラム操作	74
	メモリーの自動配分	74
	MEM	77
	キーコードとライン番号	77
	キー操作の省略	78
	プログラムの操作機能	78
	ユーザー (USER) モード	79
	プログラム・メモリー	80
	キーコードの解説	81
	プログラム作成の手順	83
	プログラムの始めと終わり	84
	完成したプログラム	85
	プログラムの入力	85
	プログラム計算	86
	USER モードでの操作	88
	プログラムの停止と一時表示	88
	プログラム計算途中の停止	89
	プログラム計算途中の一時表示	91
	計算途中の予定外の停止	93
	ラベル	94
	練習問題	94
第 6 章	プログラムの編集	96
	プログラムエラーの発見	96
	編集機能	97
	プログラム編集の例	98
	プログラムの 1 ステップずつの計算	99
	PRGM モードでの SST と BST	100
	プログラムの変更	101
	練習問題	107
第 7 章	プログラムの条件判断とコントロール	110
	プログラムの条件判断	110
	フラグ	111
	プログラムの流れのコントロール	112
	Go To	112
	ジャンプとループ	112
	フラグの使用	116

6 目 次

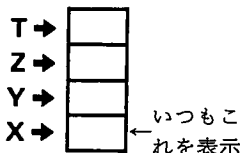
第 8 章	サブルーチン	119
	サブルーチンへのジャンプ	119
	サブルーチンの限度	120
	サブルーチンの利用法	121
第 9 章	I レジスタ	127
	I レジスタの直接指定	127
	I レジスタを使った間接指定	130
	[ISG] を使ったループ・コントロール	132
	R _I の直接指定	134
	R _I を使った間接指定	135
	プログラムの間接コントロール	136
	ジャンプとサブルーチン・ジャンプのときの	
	ラベルの間接指定	136
	ジャンプとサブルーチン・ジャンプのときの	
	ライン番号の間接指定	137
第 3 部	各種のプログラム	139
第 10 章	各種のプログラム	140
	3 × 3 マトリクス計算	140
	三元 1 次連立方程式	149
	f(x) = 0 の根 (ニュートン法)	154
	不連続点の数値積分	159
	曲線のあてはめ	162
	三角形の計算	169
	t 検定	176
	χ^2 (カイ二乗) 検定	182
	複利計算	185
	潜水艦撃沈ゲーム	194
第 11 章	プログラム作成の技術	206
	プログラムの構造	206
	問題点の把握	206
	計算手順	206
	フローチャート	208
	サブルーチン	210
	[ISG] と組合せた [RCL] (i)	211
	データの入力	212
	ループ	214
	フラグ	215
	乱数	217
	ユーザー指定キー	217

	データのストア	218
	別々のルーチンの使い分け	218
付録 A	エラー発生原因	219
付録 B	スタック上昇とラスト x	221
	数値入力の区切り	221
	スタック上昇	221
	次にスタック上昇しない操作	221
	次にスタック上昇する操作	222
	無関係な操作	222
	ラスト x	223
付録 C	メモリーの自動配分	224
	記憶レジスタからプログラム・メモリーへの変換	224
	プログラム・メモリーから記憶レジスタへの変換	227
	MEM の使い方	227
付録 D	電池, 保証と修理について	230
	電池	230
	電圧低下の症状	231
	新しい電池との交換	231
	動作の確認	233
	保証について	235
	保証の内容	235
	保証が適用されない場合	235
	修理	235
	修理料金	236
	修理完了後の保証期間	236
	修理依頼品の発送について	236
	温度範囲	237
	受信障害について	237
	プログラム手法の索引	239
	キーの索引	241
	プログラム用キーの索引	245
	事項索引	247

HP-11Cのキーと不揮発性メモリー



自動メモリースタック



ラスト x



プログラム・メモリー

専用	共用 (兼用)
000-	064-
001-	065-
002-	066-
⋮	⋮
061-	201-
062-	202-
063-	203-

数値記憶レジスタ

専用 共用 (兼用)

R ₁	<input type="text"/>	R ₀	<input type="text"/>	R ₀	<input type="text"/>
		R ₁	<input type="text"/>	R ₁	<input type="text"/>
		R ₂	<input type="text"/>	R ₂	<input type="text"/>
		R ₃	<input type="text"/>	R ₃	<input type="text"/>
		R ₄	<input type="text"/>	R ₄	<input type="text"/>
		R ₅	<input type="text"/>	R ₅	<input type="text"/>
		R ₆	<input type="text"/>	R ₆	<input type="text"/>
		R ₇	<input type="text"/>	R ₇	<input type="text"/>
		R ₈	<input type="text"/>	R ₈	<input type="text"/>
		R ₉	<input type="text"/>	R ₉	<input type="text"/>

基本のメモリー配分はプログラム用が63ライン、数値用が21レジスタです。プログラムが長いと自動的に1レジスタが7ライン分のプログラム用に変わります。1度に1レジスタずつで、R₉からR₀の順に使います。

HP-11C を 使ってみましょう

科学技術用計算機 HP-11C は簡単な計算から複雑な問題まで取扱えて、しかも重要なデータなどがスイッチを切っても消えないポケット型の優秀な計算機です。その上プログラムを組むのも使うのも簡単なのでプログラムの知識や経験は不要です。

HP-11C は電池の寿命を延ばすために、キー操作などを何もしないで 8～17分位放置すると表示が自動的に消えるようにしてあります。しかし HP-11C に入れた大切な数値などは不揮発性メモリーに記憶しているの消えてなくなることはありません。

当社の計算機はどれも他社の計算機の計算方式とは違う独特の計算方式（**ENTER** キーを使う日本語方式）を採用しています。この計算方式は少しも難かしくありません。この本で後程 **ENTER** などを詳しく説明しますが、ここで簡単な計算をして **ENTER** キーの使い方に慣れていただきましょう。

演算キーを押すと直ちに答を表示します。例えば **+** キーを押すと計算機中にあった数値にその前に入れた数値を加え、**-** キーを押せばその前に入れた数値を引き、**x** キーを押すとその前に入れた数値を掛け、**÷** キーを押すとその前に入れた数値で割ります。次のようにやってみてください。まず初めに数字キーを押して第一の数値を入れてから **ENTER** キーを押して、始めの数とその次に入れる数を区切ってやります。次に数字キーを押して第二の数値を入れ、さらに **+**、**-**、**x** または **÷** キーを押してください。

10 HP-11C を使ってみましょう

初めて計算機を使うのでしたら **ON** キーを押してスイッチを入れてください。もし 0 以外の数字を表示していたら **←** キーを押して 0.0000 と表示するようにします。小数点以下が 4 桁でなかったら、**f** **FIX** 4 と押すと以下の例題の答と同じ表示になります。(この本で特に指定がない部分は **FIX** 4 になっているものとして説明します。)



0.0000

注 スイッチを入れたときに電池が弱っていると、表示部の左下隅の星印が点滅します。そのときには付録 D (232 ページ) の通りに電池を交換してください。

手 操 作

計算を始める前に表示を 0 にしなくても結構です。しかし数字を押し間違えたときは **←** キーを押して訂正してください。

問 題	キー操作	表 示
$9 + 6 = 15$	9 ENTER 6 +	15.0000
$9 - 6 = 3$	9 ENTER 6 -	3.0000
$9 \times 6 = 54$	9 ENTER 6 x	54.0000
$9 \div 6 = 1.5$	9 ENTER 6 ÷	1.5000

上の 4 題で知っていただきたいことは次の 3 点です。

- **+** , **-** , **x** や **÷** キーを押す前に二つの数値を計算機に入れておく必要があります。
- **ENTER** キーは初めの数値と次に入れる数値を区別するために、初めの数値を入れ終ってから押します。
- 演算キー (ここでは **+** , **-** , **x** や **÷** キー) を押すと直ちに目的の計算をしてその答を表示します。

手操作とプログラムを使った計算の関係を知らいただくために、同じ問題をまず手操作で計算してみましょう。それからその問題をプログラムを使って計算してみましょう。

円筒型の温水ボイラについての計算です。 $q = hAT$ の式を使えばボイラの熱損失が簡単に計算できます。

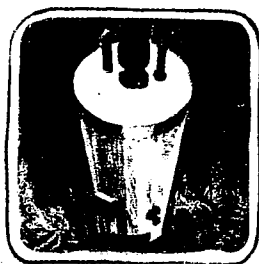
q : ボイラの熱損失 (W : ワット = ジュール/秒)

h : 熱伝達係数 (W/(m²・k))

A : ボイラの表面積 (m²)

T : ボイラの表面と周囲空気との温度差(°C)

例 200 リットル用の保温のあまり良くない円筒型の温水ボイラの熱損失を計算してみようと思います。ボイラの表面と周囲の空気との温度差を計ったら 10°C あったとします。ボイラの表面積が 3 m² で、熱伝達係数が約 1.78 として計算してみましょう。次のようにキーを押すだけで熱損失が簡単に計算できます。



キー操作	表示	
10 <input type="button" value="ENTER"/>	10.0000	温度差(T)とボイラの表面積(A)を入力。
3	3.	
<input type="button" value="x"/>	30.0000	A×T の計算。
1.78	1.78	熱伝達係数(h)。
<input type="button" value="x"/>	53.4000	熱損失(h×(AT))。
<input type="button" value="←"/>	0.0000	表示を 0 にする。

12 HP-11Cを使ってみましょう

上の例では温度差が10°Cのときの熱損失を計算しました。もし各種の温度差についての計算を何回も繰り返すときにはどうしますか？
 上のように手操作をして計算することができるようになったと思います。しかし簡単で速い方法は温度差から熱損失を計算するプログラムを作ることです。

プログラムの作成 プログラムは手操作で計算するときのキーの操作順序です。二つだけ命令を追加してあります。それはプログラムの始めと終りを区別するためのラベルと RTN です。

プログラムの入力 次のようにキー操作をして HP-11C にプログラム命令を入れてみましょう。計算機に命令を記録すれば(記憶させれば)よいのです。(キー操作したときの表示については後で説明しますので、ここでは気にしなくて結構です。)

キー操作	表示	
g P/R	000—	HP-11Cをプログラム・モードにする。(PRGM表示が見えるようになる。)
f CLEAR PRGM	000—	プログラム・メモリをクリアする。
f LBL A	001—42, 21, 11	ラベル A はプログラムの始めの目じるしです。
3	002—	} 手操作でやったときと同じキー操作。
x	003— 20	
1	004— 1	
.	005— 48	
7	006— 7	
8	007— 8	
x	008— 20	} RTN はプログラムの終りの意味。
g RTN	009— 43 32	
g P/R	0.0000	HP-11Cを計算モードに戻す。(PRGM表示が消える。)

プログラム計算の実行 次のようにキー操作をしてプログラム計算を実行します。

キー操作	表示	
10	10.	第1の温度差の入力。
f A	53.4000	手操作でやった答と同じ。
12 f A	64.0800	温度差 12°C のときの熱損失。

プログラムを入力してあればこのように種々の温度差のときの熱損失を素早く計算できます。温度差を入力してから **f** **A** と押すだけです。右表を完成させてください。

この答は 53.4000, 64.0800, 74.7600, 85.4400, 96.1200, 106.8000 になるはずです。

温度差	熱損失
10	?
12	?
14	?
16	?
18	?
20	?

プログラムはこのように簡単です。計算機はキーの操作手順を覚えていて、計算したいときにその通りに実行します。これで HP-11C の使い方が少しおわかりになったと思いますので、計算機の操作方法などをこれから詳しく説明したいと思います。

第 1 部
HP-11C の
基礎知識

第1章 まず始めましょう

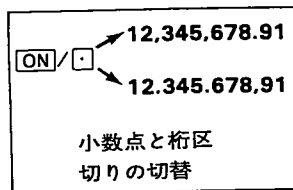
スイッチの入・切

ON キーを押すと HP-11C のスイッチを入・切できます。そして電池を長持ちさせるために、計算が終わってから 8～17 分たつと自動的にスイッチが切れるようにしてあります。

表 示

小数点と桁区切り

小数点は数値の整数部分と小数部分の区切り符号です。桁区切りは大きな数値のときに読みとりやすくするために付ける符号です。普通は小数点には点、桁区切りにコンマを付けますが、ヨーロッパでは反対に小数点にコンマ、桁区切りに点を付ける国（フランスなど）があります。HP-11C はどちらでも使えるようにしてあります。まず計算機のスイッチを切つてから、**ON** を押したまま \square キーを押す、**ON** を先にはなしてから \square もはなします（**ON**/ \square と表示します）。



状態表示

HP-11C の表示部の下半分に計算機の状態を表示する 6 個の記号があります。この表示の意味はそれぞれに応じたところで説明いたします。

* USER f g GRAD PRGM

状態表示を全部表示させたとき

負数

表示（キーを押して入力した数値や計算した答）を負数にするには **[CHS]**（change sign：符号変換）を押します。負数のときに **[CHS]** を押せば正数になります。

表示の取消し：**[CLx]** と **[←]**

HP-11Cには2種の表示取消しキーがあります。**[CLx]**（clear x：表示を0にする）と **[←]**（取消し）です。計算モードで **[9][CLx]** と押すと表示している数字が0になります。プログラム・モードで **[9][CLx]** を押すと、**[CLx]** というプログラム命令を入力したことになります。**[←]** はプログラムとして入力できない命令で、計算モードとプログラム・モードの両方で次のような取消しに使います。

1. 計算モード

- A. 計算をやった後で **[←]** を押すと表示がゼロになります。（数値を入力してから演算キーを押すと計算機は一つの数値として理解します。それから **[←]** を押すとゼロになります。）計算機の表示部の見えない部分の数字も完全に消えてしまいます。

キー操作

表示

12345

[√x]

[←]

12,345



111.1081

0.0000

計算をやった後で **[←]** を押すと全桁の数字が消えてゼロになります。

* 表示の右側の見えない部分の数字も一緒に消えてゼロになります。

18 第1章 まず始めましょう

- B. 新しい数値を入れて演算キーを押す前（例えば数値入力途中などに）を押すと、その前に入力した数字だけが取消しになります。更にを押せばその前の数字も取消しできるので完全に別な数値にすることも可能です。

キー操作

表示

12345

12,345




1,234




123



9

1,239

数値入力が終わっていないときにを押すと、数字を1文字ずつ取消することができます。

2. プログラム・モードでを押すと、表示しているプログラム命令を取消す（削除する）ことになります。

走行表示

プログラム計算を実行（走行ともいいます）中や、（permutation：順列）や（combination：組合せ）の計算中は **running** が点滅します。



running

オーバーフローとアンダーフロー

オーバーフロー 計算した結果×レジスタの絶対値が $9.999999999 \times 10^{99}$ よりも大きくなると計算不能になって、表示の全桁が9になります。（数値が負のときは負符号も表示します。）プログラム計算中にオーバーフローが発生すると、計算を中止してオーバーフローの表示になります。



9.999999 99

オーバーフローしたときの表示

アンダーフロー 計算途中で数値の絶対値が $1.000000000 \times 10^{-99}$ よりも小さくなると、その代わりにゼロになってしまいます。プログラム計算中にアンダーフローになっても計算を中止することはありません。

エラー表示

計算途中で正しくない計算（例えば負数の平方根を求めるなど）をするとエラー表示になります。

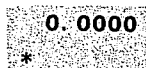
キー操作	表示
4 CHS	-4.0000
\sqrt{x}	Error 0
←	-4.0000

エラー表示とその原因は付録Cエラー原因を見てください。また HP-11C 本体の裏側にも英文の表があります。

エラー表示を消すには **←**（別のどのキーでも）を押せば、エラー発生前の状態に戻ります。

電圧低下表示

表示部の左下隅の星印が点滅したら、それは電池が弱ってきたからで、付録D (232 ページ) の通りに電池を交換してください。



メモリー

不揮発性メモリー

HP-11C は不揮発性メモリーを使っているので、スイッチを切っても次の内容は消えません。

- 計算機中の全部の数値
- 計算機に入力したプログラム
- 表示桁数の指定
- フラグのセット状態
- プログラム・メモリー中の位置
- サブルーチン実行中のリターン情報
- 角度単位（度，ラジアンまたはグラード）

20 第1章 まず始めましょう

スイッチを切る前にプログラム・モードにして (PRGM 表示が見えます) あっても、スイッチを切ってからもう一度スイッチを入れると、必ず計算モード (PRGM 表示が消えます) になります。

電池交換のために5分位電池をはずしても不揮発性メモリー内の数値やプログラムなどは消えません。

メモリーのリセット

HP-11Cの不揮発性メモリーの内容をリセット (全部消去) するには次のようにします。

1. HP-11Cのスイッチを切る。
2. **ON** キーを押したままにする。
3. **□** キーも押したままにする。
4. **ON** キーをはなしてから **□** キーもはなす。



不揮発性メモリーのリセット

メモリーをリセットすると、エラー表示に変わりますが、**←** キー (他のどのキーでも) を押せば表示が消えます。

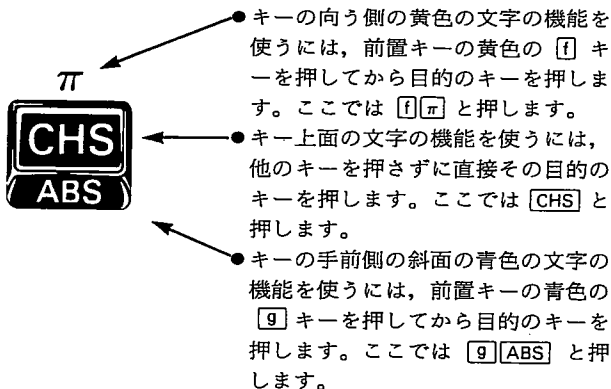


注 計算機を落したりなどすると、メモリーをリセットしなくてもメモリー内容が全部消えてしまうことがあります。

キー操作

キーの第一機能と第二機能

HP-11Cの大部分のキーは第一機能と2種の第二機能の三つを兼用しています。第一機能はキー上面の文字表示通りです。2種の第二機能はキーの向う側と、キーの手前側の斜面に文字で表示してあります。



まず **f** や **g** キーを押すと表示の下側に **f** や **g** の文字表示がついて、目的のキーを押すまで表示したままになります。

0.0000
f

前置キーの取消し

関数を計算するために前置キーを間違えて押してしまったときには、それを取消すために **f CLEAR PREFIX** と押します。(CLEAR PREFIX は **STO**, **RCL**, **GTO**, **GSB**, **HYP**, **HYP'** キーの取消しにも使えます。) この PREFIX キーは計算モードで数値の仮数部を見るためにも使います。PREFIX キーを押すと数字が10桁見え、キーをはなしてしばらくたつと普通の表示に戻ります。

単項演算キー

単項演算キーとは一つの数値だけを使って計算するキーです。次のように操作します。

1. 数値をキーインする (数値を表示していればキーインする必要はありません)。
2. 演算キー (一つだけでなく二つ以上のこともあります) を押す。

キー操作

表示

45

45

g LOG

1.6532

二項演算キー

二項演算キーとは計算機中の二つの数値を使って計算するキーです。

$\boxed{+}$, $\boxed{-}$, $\boxed{\times}$ や $\boxed{\div}$ は一番良く使う二項演算キーです。

ENTER キー 二項演算のときに片方の数値が別の計算の答なら

ENTER キーを押す必要はありません。しかしこれから二つの数値を計算機に入れるのでしたら、二つの数値を区切るために **ENTER** キーを押す必要があります。

二項演算のために計算機に二つの数値を入れるには次のようにします。

1. 初めの数値をキーインする。
2. 初めの数値と二番目の数値を区切るために **ENTER** キーを押す。
3. 二番目の数値をキーインする。
4. 二項演算キーを押す。

キー操作	表示
2	2
ENTER	2.0000
3	3
$\boxed{\div}$	0.6667

数値入力の順序 ご承知のとおり足し算と掛け算は二つの数値の順序が逆になっても答は同じです。けれども引き算と割り算は後から入れた数値を引いたり、後から入れた数値で割ります。

問題	キー操作	表示
10 - 3	10 ENTER 3 $\boxed{-}$	7.0000
3 - 10	3 ENTER 10 $\boxed{-}$	-7.0000
10 \div 3	10 ENTER 3 $\boxed{\div}$	3.3333
3 \div 10	3 ENTER 10 $\boxed{\div}$	0.3000

HP-11Cで二項演算をするときに(例えば y^x のように)キーに x と表示してあるのはいつも後から入力する数値のことです。2の3乗(2^3)を計算するには2をキーインし、**ENTER**を押してから累乗の3をキーインし、それから y^x を押します。

キー操作	表示	
2 ENTER	2.0000	
3	3	3がキーの x に相当する数値。
y^x	8.0000	$2(y)$ の $3(x)$ 乗の答。

以上の問題を自分で操作してみてください。そうすれば始めの数値をキーインしたら**ENTER**を押すということが実際にわかると思います。計算した答(または中間結果)を表示しているときには数値が自動的に区切られているので、そのまま次の数値をキーインすることができます。

$(2+4) \div 8$ を求めるには

キー操作	表示	
2 ENTER	2.0000	
4	4	
+	6.0000	$(2+4)$
8	8	
÷	0.7500	$(2+4) \div 8$

$(9+17-4+23) \div 4$ を求めるには

キー操作	表示	
9 ENTER	9.0000	
17 +	26.0000	$(9+17)$
4 -	22.0000	$(9+17-4)$
23 +	45.0000	$(9+17-4+23)$
4 ÷	11.2500	$(9+17-4+23) \div 4$

もっと複雑な計算でも中間結果を自動的に記憶していますので、同じように簡単に計算できます。

24 第1章 まず始めましょう

例 $(6+7) \times (9-3)$ を計算しましょう。

まず $(6+7)$ の中間結果を求めます。

キー操作	表示
6	6
ENTER	6.0000
7	7
+	13.0000 (6+7)

次に $(9-3)$ を計算する番です。中間結果のほかに別の二つの数値を入力することになるので、始めの数 (9) をキーインして **ENTER** を押し、次の数 (3) をキーインします。(計算機は中間結果の 13 を表示しているので、9 をキーインする前には **ENTER** を押す必要はありません。中間結果は自動的に内部に記憶しています。)

キー操作	表示
9	9
ENTER	9.0000
3	3
-	6.0000 (9-3)

それから中間結果どうし(13と6)を掛けて答を求めましょう。

キー操作	表示
x	78.0000 $(6+7) \times (9-3) = 78$

このように HP-11C は中間結果を自動的に記憶していて、後入れ先出しの原則で一つずつ計算します。複雑そうに早える問題も順番に分解すれば単項演算や二項演算を続けたものになります。

覚えておいていただきたいこと

- 二つの数値を続けてキー・インするときに、始めの数値をキー・インしたら次の数値と区切るために **ENTER** キーを押します。
- 計算した後ではどんな数をキー・インしても前の答とは別なものと計算機が判断します。

- 中間結果は後入れ先出しの原則で記憶しています。

それでは次の問題をやってみてください。筆算でする順番どりに計算をすればよいのです。HP-11Cは中間結果を自動的に覚えているので順番どりにすれば間違えることはありません。

$$(16 \times 38) - (13 \times 11) = 465.0000$$

$$(27 + 63) \div (33 \times 9) = 0.3030$$

$$(\sqrt{(16.38 \times 0.55)}) \div .05 = 60.0300$$

$$4 \times (17 - 12) \div (10 - 5) = 4.0000$$

第2章 自動メモリー・スタック, LAST_x, 数値のストア

自動メモリー・スタックとスタック操作

HP-11C は中間結果を自動的に記憶していてすぐに使えるので、複雑な計算も簡単な操作でできます。これは自動メモリー・スタックと

ENTER キーを利用しているからです。

自動メモリー・スタック

T →	0.0000
Z →	0.0000
Y →	0.0000
X →	0.0000

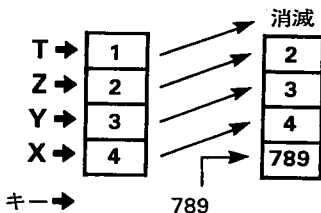
いつもこれを表示します。

HP-11C は計算モード（表示に PRGM の文字が見えないとき）ではいつも X レジスタの数値を表示します。

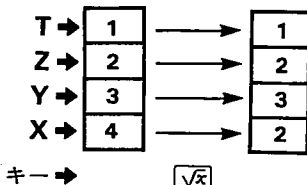
キーインした数値や計算結果は必ず X レジスタに入るのでこれが見られます。この表示する数値はキーインした数や計算結果だけでなく、スタック操作などの結果によっても変わります。スタックに入れた数値は必ず“後入れ先出し”で使っています。スタック内に次のような数値が入っているとして（例えば前の計算結果などが）、図のようにキーを押すとスタック内の数値が左から右のように変化します*。

* この章の説明図ではわかりやすくするために、この本で普通に使っている表示形式（**FIX** 4）ではなく大部分は1桁の数字だけにしてあります。

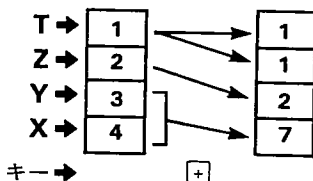
スタック上昇



スタック移動なし

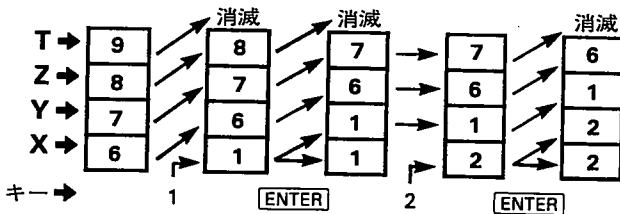


スタック下降

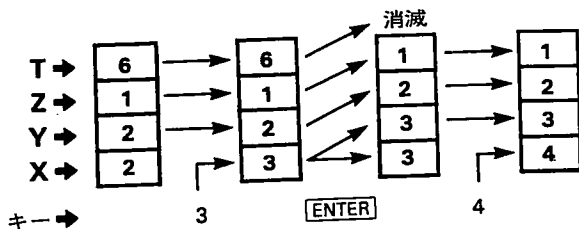


スタック操作

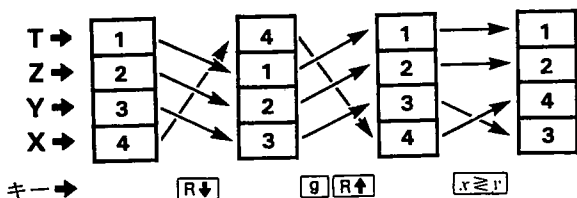
ENTER はある数と別の数を区切るのに使いました。この **ENTER** を押すと図のように Xレジスタに入っていた数値（それまで表示していた数値）が Yレジスタにもコピーして入ります。例えばスタックに 1, 2, 3, 4 を入れる（それ以前の計算結果がスタック内に入っていると仮定して説明します）と図のように変わります。



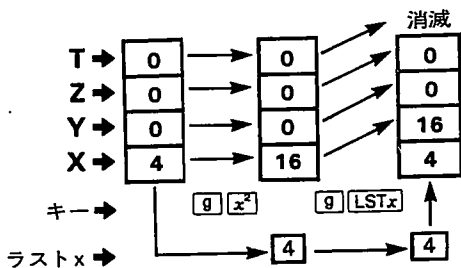
(次のページに続きます)



$\boxed{R\downarrow}$ (roll down : ロールダウン), $\boxed{R\uparrow}$ (roll up : ロールアップ) と $\boxed{x\rightleftharpoons y}$ (x change y : x と y の入替え) $\boxed{R\downarrow}$ と $\boxed{R\uparrow}$ はスタック内の数値を下方または上方に移動させるキーです。このときには数値の消滅はありません。 $\boxed{x\rightleftharpoons y}$ はXレジスタとYレジスタ内の数値の入替え用です。スタック内に図のように1, 2, 3, 4が入っているときに $\boxed{R\downarrow}$, $\boxed{g R\uparrow}$ や $\boxed{x\rightleftharpoons y}$ を押すと図のようになります。



\boxed{LSTx} (last x : ラスト x) 演算キーを押したときにはその直前に表示していたXレジスタの数値をラスト x レジスタにストア (記憶) しています。 $\boxed{g LSTx}$ を押すとそのときにラスト x レジスタに入っている数値をXレジスタにコピーして表示します。(付録B スタック上昇とラスト x にどんな関数のときに x の値がラスト x レジスタに入るかの表をまとめておきました。) 下図はスタック内の数値の移動例です。

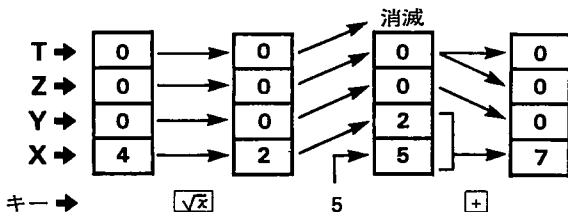


関数の計算とスタック

二つの数値をキーインするときには一つの数値をキーインし、**ENTER**を押してから二つ目の数値をキーインします。計算した答やスタック内移動 (**x \geq y**) や **R \uparrow** など) の操作した結果を表示しているときにはもう一つの数値をキーインするときには前もって **ENTER** を押す必要はありません。なぜそうなるのでしょうか? それは下記の4種以外は二つの役割りを持たせているからです。

1. 目的の本来の機能 (関数計算など) を実行する。
2. 自動メモリー・スタックを可能にする。これはその後で新しい数値をキーインするとそれまでXレジスタなどに入っている数値が上昇するようになることです。

ここではXレジスタに4が入っているときに次のキー操作したときの説明図を見てください。



例外の4種は **ENTER**, **CLx**, **$\Sigma+$** と **$\Sigma-$** で自動メモリー・スタックは働きません*。つまりこのキーの操作後に次の数値をキーインしてもY~Tレジスタの数値は動きません。このキー操作後に次の数値をキーインすると、それまでXレジスタに入って表示していた数値が新しい数値に切替わるだけです。(**ENTER** を押すとそのときにスタック上昇をしますが、その後で別の数値を入れてもスタック上昇はしません。この **ENTER** については27~28ページの説明図を見てください。) 普通の計算をやっているときにはこのことをいちいち考えなくても大丈夫です。

* 数値をキーインして他のキーを押した後に **+** を押すと **CLx** のときと同じようにスタック上昇はしません。数値のキーイン途中ではスタック上昇に関係ないキーです。詳しいことは付録Bスタック上昇とラストxを見てください。

二項演算

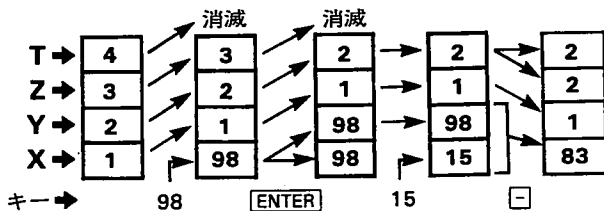
二項演算で大切なことはスタックに入っている数値の位置です。加減乗除のときには紙に筆算するときの順序で数値をキーインする必要があります。例えば 98 から 15 を引くときにはつぎのようにまず 98 を書きその下に 15 を書きますね。

$$\begin{array}{r} 98 \\ -15 \\ \hline \end{array}$$

筆算では次に下のように引き算をします。

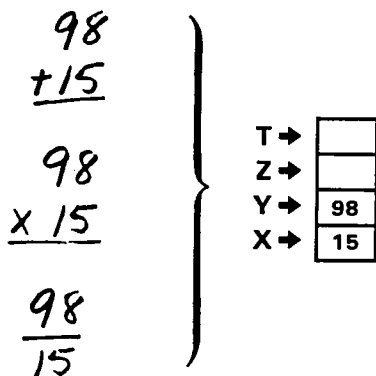
$$\begin{array}{r} 98 \\ -15 \\ \hline 83 \end{array}$$

HP-11Cで計算するときも同じ順序でキーインします。つまり元の数(引かれる数)をYレジスタに入れ、第二の数(引く数)をXレジスタに入れて表示させます。それから \square キーを押すとYレジスタの98からXレジスタの15を引きその答はXレジスタに入り、Z~Tの値がスタック下降します。この計算のときのスタック内の動きは下図の通りです。(ここではスタック内に前の計算結果が入っているものとして説明します。)



普通の計算はどんなときでも筆算と同じ計算順序でキーインして計算するとスタック下降を生じます。ここでは 98 から 15 を引く引き算を

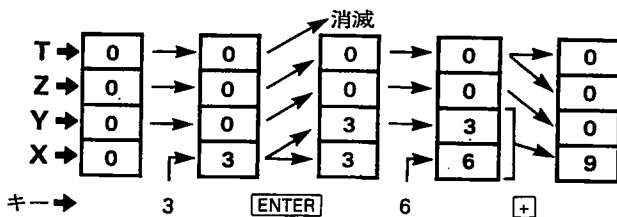
やりましたが, 同じように 98 に 15 を足したり, 98 に 15 を掛けたり, 98 を 15 で割ったりしてください。

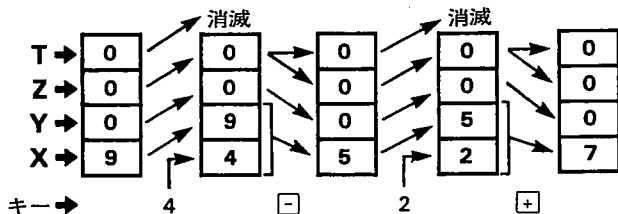


長い計算

HP-11C を使うのは直接演算キーを押す手操作か, プログラムを組んでおいてから計算のどちらかだと思いますが, 連続計算がかなりあると思います。そこでこれから HP-11C の計算方法がどんなに優れているのか説明しましょう。どんなに複雑な長い計算でも一回ごとに一つの計算の繰返しです。自動メモリー・スタックに最大4個までの中間結果を入れておき, 必要になったらそれを使って計算すればよいわけです。連続計算のときはどうなるか見るために $3+6-4+2=$ の計算をやってみましょう。(スタック内をクリアするには \leftarrow [ENTER])

[ENTER] [ENTER] とキーを押してください。





このように1回ごとに一つの計算をやりました。スタックに二つの数値を入れておいて計算するとスタック下降をします。その次に新しい数値をキーインすると自動的にスタック上昇をします。もっと複雑な計算でも一つ一つに分解するとこのように簡単になります。

例 これまではスタック内の数値の移動を矢印で表示していましたが、これからは次式のような計算には下表の形式で表示します。

$$\frac{(3+4) \times (6-4)}{2}$$

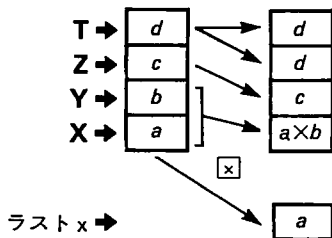
T →	0	0	0	0	0	0
Z →	0	0	0	0	0	7
Y →	0	3	3	0	7	6
X →	3	3	4	7	6	6
キー →	3	ENTER	4	+	6	ENTER

T →	0	0	0	0	0
Z →	7	0	0	0	0
Y →	6	7	0	14	0
X →	4	2	14	2	7
キー →	4	-	×	2	÷

ラスト x

HP-11Cのラストxレジスタは普通のは別のレジスタ（数値を記憶しておくところ）で、演算をする前に表示していた値を一時的に記憶

させておくところです。



これは計算に使った値をまた使うときや、計算を間違えた（キーを押し間違えた）ときに同じ数値をもう一度キーインし直さなくて済ませるためです。

例 45.575mと25.331mのように二つの別々の数値に同じ数値（ここでは0.175）を掛けてみましょう。

T →	0.0000	0.0000	0.0000	0.0000
Z →	0.0000	0.0000	0.0000	0.0000
Y →	0.0000	45.5750	45.5750	0.0000
X →	45.575	45.5750	0.175	7.9756

キー → 45 [.] 575 [ENTER] [.] 175 [x]

ラストx → [0.1750]

T →	0.0000	0.0000	0.0000
Z →	0.0000	7.9756	0.0000
Y →	7.9756	25.3310	7.9756
X →	25.331	0.1750	4.4329

キー → 25.331 [g] [LSTx] [x]

ラストx → [0.1750] [0.1750] [0.1750]

* 例外は統計計算の [Σ], [S] と [L.R.] です。

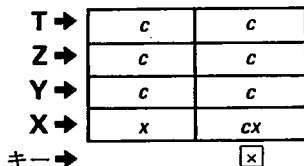
34 第2章 自動メモリー・スタック, LASTx, 数値のストア

LSTx を使うとキーの操作ミス（演算キーの押し違いや数値のキーイン・ミス）を簡単に修正できます。例えば 287 を 13.9 で割るときに間違えて 12.9 で割ったときの修正法をお目にかけてみます。

キー操作	表示	
287 ENTER	287.0000	
12.9 +	22.2481	しまった！ 数字の押し間違いだった。
9 LSTx	12.9000	+ を押す前の Xレジスタの値（正しくない値）をラスト x レジスタから呼び出す。
x	287.0000	押したキーとは逆の計算キーを押して元の値に戻す。
13.9 ÷	20.6475	正しい答。

定数計算

Tレジスタに入っている値はスタック下降をしてもそのまま残っているので、これを利用して定数計算ができます。



この計算をするには、まず定数をキーインして X レジスタに入れ **ENTER** を 3 回押して T レジスタに入れます。次にこの定数を使って計算したい数をキーインし、演算キーを押します。演算キーを押すたびにスタック下降がおきて T レジスタの値がコピーされるので次の定数計算ができます。

例 ある細菌学者が1日の増殖率が15% (増殖比は1.15) の細菌の増殖試験を計画しています。1000個からスタートしたら5日後までの毎日の細菌数は幾つになるでしょうか?



方法 **ENTER** を使って増殖比 (1.15) を Y, Z, Tレジスタまで入れ、次にスタート数 (1000) を Xレジスタに入れます。それから **x** を押すたびに翌日の細菌数が計算できます。**f** **FIX** 2 を押して表示形式を下の図のようにしてください。

T →	0.00	0.00	0.00	1.15	1.15
Z →	0.00	0.00	1.15	1.15	1.15
Y →	0.00	1.15	1.15	1.15	1.15
X →	1.15	1.15	1.15	1.15	1,000

キー → 1.15 **ENTER** **ENTER** **ENTER** 1000

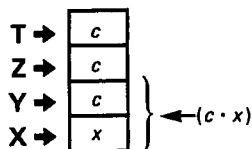
T →	1.15	1.15	1.15	1.15	1.15
Z →	1.15	1.15	1.15	1.15	1.15
Y →	1.15	1.15	1.15	1.15	1.15
X →	1,150.00	1,322.50	1,520.88	1,749.01	2,011.36

キー → **x** **x** **x** **x** **x**

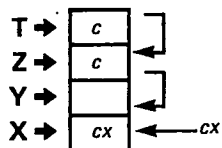
最初に **x** を押すと 1.15×1000 の計算をします。この答 (1,150.00) が Xレジスタに入って表示され、スタック下降をして、Tレジスタに増

増殖比のコピーが入ります。 \boxed{x} を押すたびにこれを繰り返します。

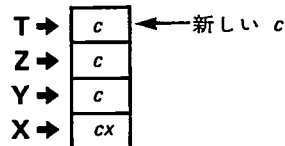
1. XとYレジスタの値を使って計算します。



2. その計算結果がXレジスタに入って表示され、スタック内の値が一つずつ下に下がる。



3. それまでTレジスタにあった値 (ここでは増殖比)のコピーがTレジスタに入る。



このようにスタック下降がおきるたびにそれまでTレジスタに入っていた増殖比のコピーが入るので、キーインし直す必要がないわけです。

\boxed{f} \boxed{FIX} 4 を押して HP-11C 本来の \boxed{FIX} 4 表示に戻してください。

別の方法 ラスト x レジスタを使っても定数計算ができます。この方法で上と同じ計算をやってみてください。

1. スタート数 (1,000) をキーインして \boxed{ENTER} を押す。
2. 増殖比 (1.15) をキーインする。
3. \boxed{x} を押して翌日の細菌数を計算する。
4. \boxed{g} \boxed{LSTx} \boxed{x} と押してその翌日の細菌数を計算する。

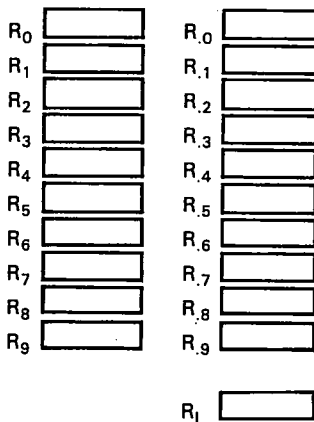
記憶レジスタの操作

表示しているXレジスタの値を記憶させたり、それを呼出して利用するために HP-11C には 21 個の記憶レジスタ (いわゆるメモリー) があります。この記憶レジスタはスタックやラスト x レジスタとは完全に別になっています。

数値の記憶 (ストア)

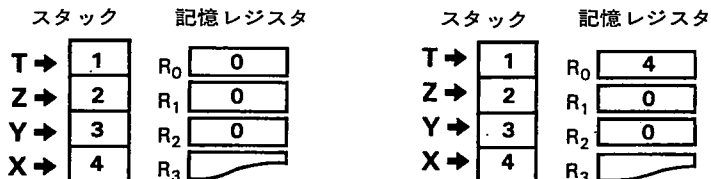
[STO] (store) このキーを押してから記憶レジスタの番号 (0 から 9, \square 0 から \square 9 または \square) を押すと表示している X レジスタの値を指定したレジスタに記憶できます。

記憶レジスタ



このときに

[STO]0 と押すと



38 第2章 自動メモリー・スタック, LASTx, 数値のストア

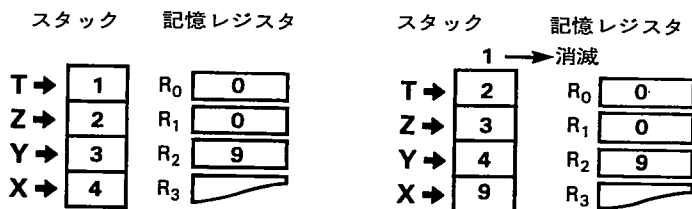
記憶レジスタに記憶させた値はその記憶レジスタに別の値を記憶させるか、記憶レジスタ全体をクリアするか、不揮発性メモリーをリセットするまで記憶し続けます。

記憶させた数値の呼出し (リコール)

[RCL] (recall) **[RCL]** を押してからレジスタの番号 (0 から 9, \square 0 から \square 9, または \square) を押すとその記憶レジスタに入っている (記憶させた) 数値のコピーを Xレジスタに入れて表示します。そのときスタック上昇をしない状態 (その前に **[ENTER]**, **[CLx]**, (\leftarrow), **[$\Sigma+$]**, **[$\Sigma-$]** を押したとき) でなければ **[RCL]** するとスタック上昇をします。

このときに

[RCL]2 と押すと



ストアとリコールの練習

次の練習をやってみてください。

キー操作	表示	
123	123	
[STO] 4	123.0000	R ₄ に 123 を記憶させる。
678	678	
[STO] \square 7	678.0000	R ₇ に 678 を記憶させる。
[RCL] 4	123.0000	R ₄ から 123 を呼出す。
[RCL] \square 7	678.0000	R ₇ から 678 を呼出す。

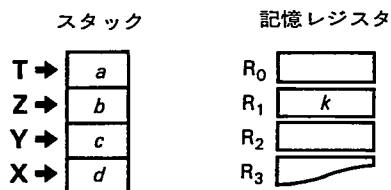
記憶レジスタのクリア

CLEAR **[REG]** (clear registers) **[f]** **CLEAR** **[REG]** と押すと記憶レジスタ全体の値が消えて全部 0 になります。このキーを押してもスタ

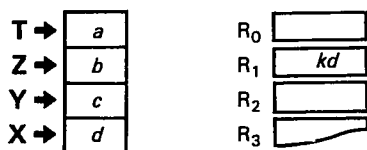
ックやラストxレジスタの数値には関係ありません。

記憶レジスタの直接四則演算

[STO] (**[+]**, **[-]**, **[x]**, **[÷]**) n 表示している X レジスタの値を使って Rn の値との直接四則演算をすることができます。キーを押す順序は **[STO]** を押し、次に四則演算キーを押し、それからレジスタ番号 (0~9)* を押します。直接四則演算の結果はその記憶レジスタに記憶します。



スタックと記憶レジスタが上のようなときに **[STO]** **[x]** 1 と押すと下のようになります。



(スタックは変化しない)

R₁ の値が k から (k × d) に変化した

* R₀ から R₉ までの記憶レジスタの直接四則演算は間接指定をしないとできませんので第9章1レジスタを見てください。また R_i との直接四則演算は出来ません。

40 第2章 自動メモリー・スタック, LASTx, 数値のストア

記憶レジスタとの直接四則演算の練習

キー操作

表示

18 [STO] 0	18.0000	R ₀ に 18 をストア。
3 [STO] [÷] 0	3.0000	R ₀ の値 (18) を 3 で割る。
[RCL] 0	6.0000	新しい R ₀ の値を呼出す。
4 [STO] [×] 0	4.0000	新しい R ₀ の値 (6) に 4 を掛ける。
[RCL] 0	24.0000	新しい R ₀ の値を呼出す。
[STO] [+] 0	24.0000	新しい R ₀ の値に 24 をたす。
[RCL] 0	48.0000	新しい R ₀ の値を呼出す。
40 [STO] [−] 0	40.0000	新しい R ₀ の値 から 40 を引く。
[RCL] 0	8.0000	新しい R ₀ の値を呼出す。

練習問題

1. 次の式の x を求めてください。

$$x = \sqrt{\frac{8.33(4 - 5.2) \div ((8.33 - 7.46) \times 0.32)}{4.3(3.15 - 2.75) - (1.71 \times 2.01)}}$$

答 4.5728

キー操作順序の一例 (この順序でなくても答を求められます)

```

4 [ENTER] 5.2 [−]
8.33 [×] [9] [LSTx] 7.46
[−] .32 [×] [÷]
3.15 [ENTER] 2.75 [−]
4.3 [×] 1.71 [ENTER]
2.01 [×] [−] [÷] [√x]
    
```

2. 定数計算の方法で、借入額 1000 ドルのローンで毎月 100 ドルずつ返したときの半年後の借入残額を計算してください。ただし月利 i は 1% とします。

計算法 計算式は次の通りなので、スタック全体に $(1+i)$ を入れ、それから借入額をキーインします。式の通りに計算すると返済後の借入残額が計算できます。

$$\text{新しい借入残額} = ((\text{前回の残額}) \times (1+i)) - \text{返済額}$$

答 446.3186 ドル

3. R₅ に 100 をストアしてから次の順序で計算してください。

1. R₅ の内容を 25 で割る。
2. R₅ の内容から 2 を引く。
3. R₅ の内容に 0.75 を掛ける。
4. R₅ の内容に 1.75 を足す。
5. R₅ の内容を呼出す。

答 3.2500

第 3 章 関数

HP-11C には数値変更用、一般数学用や統計用の各種の関数を内蔵させました。どの関数も手操作でき、しかもプログラム命令にも使えます。

π (Pi: 円周率)

f **π** キーを押すと Xレジスタに π の 10 桁の近似値 (3.141592654) が入ってこれを表示します。キーを押す前がスタック上昇をしない状態でなければ、キーを押すとスタック上昇をします。

数値変更用

HP-11C には **CHS** (change sign: 符号変換, 17 ページ参照) 以外に、次の 4 種の数値変更の関数があります。**ABS**, **INT**, **FRAC** と **RND** です。

絶対値 **g** **ABS** (absolute value) キーを押すと、Xレジスタにある数値をその絶対値に変更 (負符号だけを取る) します。

整数部取出し **g** **INT** (integer portion) キーを押すと、Xレジスタにある数値をその整数部分だけを取り出し (小数部分を切捨てて小数点以下を全部ゼロにする) します。

小数部取出し **f** **FRAC** (fractional portion) キーを押すと、Xレジスタにある数値をその小数部分だけを取り出し (整数部分を切捨てて整数部分をゼロにする) します。

四捨五入 **g** **RND** (rounding) キーを押すと、Xレジスタにある 10 桁の仮数をそのときの表示指定 (**FIX**, **SCI** や **ENG**) つまり表示しているとうりの数値に四捨五入します。

計算目的	キー操作例	表示
絶対値	12345 [CHS]	-12.345
整数部取出し	[g] [ABS] 123.4567	12.345.0000 123.4567
小数部取出し	[g] [INT] 123.4567	123.0000 123.4567
四捨五入 ([FIX] 4 にセッ トしてあるとき)	[f] [FRAC] 1.23456789	0.4567 1.23456789
中身チェック	[g] [RND]	1.2346
[FIX] 4. に戻す	[f] [FIX] 8	1.23460000
	[f] [FIX] 4	1.2346

単項演算関数

HP-11Cの単項演算関数はどれも次のような性質です。

- そのときに表示しているXレジスタの数値を関数計算に使う。
- Xレジスタの数値を使って関数計算した結果がXレジスタに入って(計算前の数値は消えます)それを表示します。
- Y, Z, Tレジスタの数値には何の影響もありません。

普通関数

逆数 **[1/x]** キーを押すと、そのときに表示しているXレジスタの数値の逆数(つまり1をその数で割った答)を計算して表示します。

階乗とガンマ関数 **[f]** **[x!]** キーを押すと、次の階乗またはガンマ関数を計算します。

1. 階乗 (factorial)

そのときに表示しているXレジスタの数値 n が正の整数 ($0 \leq n \leq 69$) であれば n の階乗(つまり1から n までを全部掛合せた答)を計算して表示します。

44 第3章 関数

2. ガンマ関数 (gamma function)

[x] キーは記号 $\Gamma(x)$ のガンマ関数 (高等数学や統計で使います)*も計算できます。 **[x]** を押すと $\Gamma(x+1)$ が求められますので、或る数のガンマ関数を求めるにはその数から1を引きます。その答を表示したら **f [x]** を押します。

平方根 **[√x]** (square root) キーを押すと、そのときに表示している Xレジスタの数値の平方根を計算します。

二乗 **[g x²]** (square) キーを押すと、そのときに表示している Xレジスタの数値の二乗を計算します。

計算目的	キー操作例	表示
逆数	25 [1/x]	25 0.0400
階乗	8 f [x]	8 40,320.0000
ガンマ関数	2.7 [ENTER] 1 [-] f [x]	2.7 1.7000 1.5447
平方根	3.9 [√x]	3.9 1.9748
二乗	12.3 [g x²]	12.3 151.2900

• 階乗とガンマ関数の関係は x が正の整数 n であれば $\Gamma(x+1) = \Gamma(n+1) = n!$ です。ガンマ関数は階乗のように正の整数のときだけという制限がありません。そこで階乗はガンマ関数の特別な状態と考えるとよいでしょう。負の整数についてはガンマ関数の定義がありませんので、負の整数のときに **f [x]** を押すと負のオーバーフローになります。

三角関数

6種の三角関数の計算の前に角度の単位を指定してください。

角度の単位 角度の単位を指定しないと正しい値の計算ができないことがあります。三角関数の計算の前に計算機の角度単位（度，ラジアン，グラード）の表示がどうか見てください。

度単位 **[g] [DEG]** キーを押すと度単位に切替わります。角度単位の表示はありません。

ラジアン単位 **[g] [RAD]** キーを押すとラジアン単位に切替わります。ラジアン単位になると表示に RAD の表示が見えます。

[g] [RAD]

0.0000
RAD

グラード単位 **[g] [GRD]** キーを押すとグラード単位に切替わります。グラード単位になると表示に GRAD の表示が見えます。

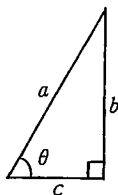
[g] [GRD]

0.0000
GRAD

計算機はいつも以上3種のうちのどれかになっています。不揮発性メモリーが最後の角度指定を記憶しているので、スイッチを切ってもスイッチを入れれば、スイッチを切る前の状態に戻ります。電池がだめになったり、不揮発性メモリーをリセット（20 ページ参照）すると自動的に **[DEG]** 単位になります。

三角関数

キー	計算
[SIN]	サイン（角度が θ のときの $\frac{b}{a}$ ）
[g] [SIN⁻¹]	アークサイン（ $\frac{b}{a}$ のときの角度 θ ）
[COS]	コサイン（角度が θ のときの $\frac{c}{a}$ ）
[g] [COS⁻¹]	アークコサイン（ $\frac{c}{a}$ のときの角度 θ ）
[TAN]	タンジェント（角度が θ のときの $\frac{b}{c}$ ）
[g] [TAN⁻¹]	アークタンジェント（ $\frac{b}{c}$ のときの角度 θ ）



三角関数を計算するときには角度の単位 ($\boxed{\text{DEG}}$, $\boxed{\text{RAD}}$ または $\boxed{\text{GRD}}$) がどれになっているか必ず確かめてください。

($\boxed{\text{SIN}}$, $\boxed{\text{COS}}$, $\boxed{\text{TAN}}$ を計算するときの値は必ず 10 進法を使い、 $\boxed{\text{SIN}^{-1}}$, $\boxed{\text{COS}^{-1}}$, $\boxed{\text{TAN}^{-1}}$ の答は必ず 10 進法になります。)

計算目的	キー操作例	表示
(操作例は $\boxed{\text{DEG}}$ のときです。)		
三角関数はどれでも 例えば サイン	33.5 $\boxed{\text{SIN}}$	33.5 0.5519
アークサイン	.7982 $\boxed{9} \boxed{\text{SIN}^{-1}}$	0.7982 52.9586

時間と角度の変換

HP-11C は時間と角度を 10 進法と 60 進法 (分と秒を使って表わす) のどちらでも扱え、互いに変換できます。

10 進法の時間 H.h または 10 進法の度 D.d	60 進法の時間 H.MMSSs または 60 進法の度 D.MMSSs
------------------------------------------	--------------------------------------------------

H が時間の整数部分
h が時間の小数部分
MM が分 (0 ~ 59)
SS が秒 (0 ~ 59)
s が秒未満の小数部分
D が度の整数部分
d が度の小数部分

60 進法への変換 $\boxed{f} \boxed{\rightarrow\text{H.MS}}$ キーを押すと表示している X レジスタの 10 進法の時間 (または度) を 60 進法の時間 (または度) ・分 ・秒に変換します。

H.h → H.MMSSs
または
D.d → D.MMSSs

10進法への変換 \boxed{g} $\boxed{\rightarrow H}$ キーを押すと表示しているXレジスタの時間（または度）・分・秒を10進法の時間（または度）に変換します。

H.MMSSs - H.h
または
D.MMSSs - D.d

度とラジアン相互変換

$\boxed{\rightarrow DEG}$ と $\boxed{\rightarrow RAD}$ 関数は10進法の度とラジアンの相互変換 (D.d \rightarrow R.r と R.r \rightarrow D.d) に使います。

度からラジアンへの変換 \boxed{f} $\boxed{\rightarrow RAD}$ キーを押すと表示しているXレジスタの度数を同じ角度のラジアン数に変換します。

ラジアンから度への変換 \boxed{g} $\boxed{\rightarrow DEG}$ キーを押すと表示しているXレジスタのラジアン数を同じ角度の度数に変換します。

計算目的	キー操作例	表示
10進法の時間 (H.h) または度 (D.d) を60進法の時間 (H.MMSSs) または度 (D.MMSSs) に変換	17.553 \boxed{f} $\boxed{\rightarrow H.MS}$	17.553 17.3311
\boxed{FIX} 4以下の部分を見る ときには	\boxed{f} \boxed{PREFIX}	1733108000
H.MMSSs または D.MMSSs を10進法の時間 (H.h) または度 (D.d) に変換	12.3045 \boxed{g} $\boxed{\rightarrow H}$	17.3311 12.3045 12.5125
度からラジアンへ	40.5 \boxed{f} $\boxed{\rightarrow RAD}$	40.5 0.7069
ラジアンから度へ	1.1746 \boxed{g} $\boxed{\rightarrow DEG}$	1.1746 67.2996

対数

自然対数 **[g] [LN]** キーを押すと表示している Xレジスタの値の自然対数 (natural logarithm: $\log_e x$ ただし e は 2.718281828) を計算します。

逆自然対数 **[e^x]** キーを押すと表示している Xレジスタの値の自然対数の逆関数 (つまり e^x ただし e は 2.718281828) を計算します。

常用対数 **[g] [LOG]** キーを押すと表示している Xレジスタの値の常用対数 (common logarithm: $\log_{10} x$) を計算します。

逆常用対数 **[10^x]** キーを押すと表示している Xレジスタの値の常用対数の逆関数 (つまり 10^x) を計算します。

計算目的	キー操作例	表示
自然対数	45 [g] [LN]	45 3.8067
逆自然対数	3.4012 [e^x]	3.4012 30.0001
常用対数	12.4578 [g] [LOG]	12.4578 1.0954
逆常用対数	3.1354 [10^x]	3.1354 1,365.8405

双曲線関数 (hyperbolic functions)

キー操作	計 算
[f] [HYP] [SIN]	ハイパーボリック・サイン $\left(\sinh x = \frac{e^x - e^{-x}}{2} \right)$
[g] [HYP⁻¹] [SIN]	逆ハイパーボリック・サイン $\left(\sinh^{-1} x = \ln(x + \sqrt{x^2 + 1}) \right)$

キー操作	計算
f [HYP] [COS]	ハイパーボリック・コサイン $\left(\cosh x = \frac{e^x + e^{-x}}{2}\right)$
g [HYP¹] [COS]	逆ハイパーボリック・コサイン $\left(\cosh^{-1} x = \ln(x + \sqrt{x^2 - 1})\right) \text{ただし } x \geq 1$
f [HYP] [TAN]	ハイパーボリック・タンジェント $\left(\tanh x = \frac{e^x - e^{-x}}{e^x + e^{-x}}\right)$
g [HYP¹] [TAN]	逆ハイパーボリック・タンジェント $\left(\tanh^{-1} x = \sqrt{\ln\left(\frac{1+x}{1-x}\right)}\right) \text{ただし } x^2 < 1$

計算目的	キー操作例	表示
双曲線関数はどれも 例えば ハイパーボリック・サイン	2.53 f [HYP] [SIN]	2.53 6.2369
逆ハイパーボリック・サイン	1.95 g [HYP¹] [SIN]	1.95 1.4210

二項演算関数

HP-11Cの二項演算は表示しているXレジスタの数値と、Yレジスタの数値（それ以前に入れた数や中間結果など）の両方を使って計算します。二つの数をキーインして計算するには、まずYレジスタに入れる数値をキーインしてから**[ENTER]**を押してYレジスタに入れ、次の数をキーインして表示させてから演算キーを押します。

累乗（指数乗、べき乗）

[x^y]キーを押すとYレジスタ数値のXレジスタ数値乗を計算します。

計算目的	キー操作例	表示
累乗	2 [ENTER]	2.0000
	3	3
	[x^y]	8.0000

パーセント

パーセント ある数の何パーセントを計算するには

1. 基準の数をキーインする。
2. **[ENTER]** を押す。
3. パーセント比をキーインする。
4. **[g][%]** キーを押す。

T →				
Z →				
Y →		150	150	150
X →	150	150	25	37.5

(y)

(x × y)

キー → 150 **[ENTER]** 25 **[g][%]**

ラストX →

[25]

答はXレジスタに入って表示しますが、基準の数はYレジスタにそのまま残り、パーセント比の数はラストXレジスタに入ります。**[g][%]** を押す前にZとTレジスタにある数値はそのまま、スタック内の移動はありません。上の図は150の25%を計算したときの状態を説明したものです。

増減率(%) **[Δ%]** キーを押すと、基準数(始めの数)から第二の数(増減後の数)への増減率(%)を計算します。次のように操作します。

1. 基準の数(増減前の数)をキーインする。
2. **[ENTER]** を押す。
3. 第二の数(増減後の数)をキーインする。
4. **[g][Δ%]** を押す。

T →				
Z →				
Y →		150	150	150
X →	150	150	225	50

(y)

 $\left(\frac{x-y}{y} \times 100\right)$

キー → 150 **[ENTER]** 225 **[g][Δ%]**

ラストX →

[225]

この順序で計算すると、基準の数よりも第二の数の方が増えていれば

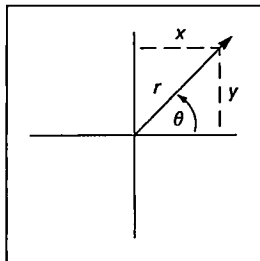
正の数、第二の数が減っていれば負の数になります。上図では150(基準の数)が225(第二の数)に増えたときの増減率を計算状態を説明したものです。

計算目的	キー操作例	表示
パーセント 基準の数	200 [ENTER]	200.0000
	パーセント比	75
	答	150.0000
増減率 基準の数	40 [ENTER]	40.0000
	増減後の数	160
	答	300.0000

極座標と直交座標の変換

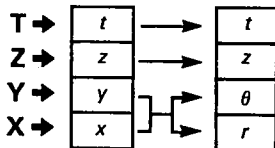
HP-11Cには極座標と直交座標の座標系換算用に $\rightarrow P$ と $\rightarrow R$ の二つのキーを用意してあります。

右図の角度 θ はそのときの計算機の角度の単位 (DEG , RAD または GRD) で測った10進法の角度とします。図のように+方向の x 軸からの角度で測り、答は $+180^\circ$ と -180° の間の角になります。



直交座標から極座標へ

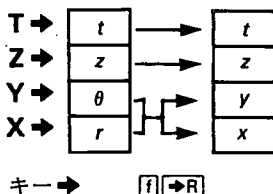
$\text{g} \rightarrow P$ (polar) キーを押すと X と Y レジスタにある直交座標系の座標値 (x , y) を極座標系の座標値 (r , θ) に換算します。



キー \rightarrow $\text{g} \rightarrow P$

極座標から直交座標へ

f **↔R** (rectangular) キーを押すと X と Y レジスタにある極座標系の座標値 (r, θ) を直交座標系の座標値 (x, y) に換算します。



計算目的		キー操作例	表示
直交座標から極座標へ	y	5 ENTER	5.0000
	x	10	10
	r	g ↔P	11.1803
	θ	x ≧ y	26.5651
極座標から直交座標へ	θ	30 ENTER	30.0000
	r	12	12
	x	f ↔R	10.3923
	y	x ≧ y	6.0000

$$r = \sqrt{x^2 + y^2} \quad \theta = \tan^{-1} \frac{y}{x} \quad x = r \cos \theta \quad y = r \sin \theta$$

順列と組合せ

順列 **f** **P_{y,x}** キーを押すと、 y 個の異なるものから x 個取出して並べるときの並べ方の方法 (つまり順列) を計算します。 (x を変えて計算するときには y も入れ直す必要があります。) 計算式は次の通りです。

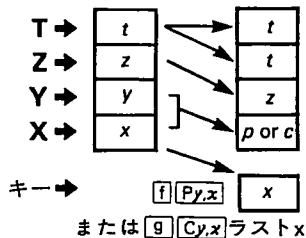
$$P_{y,x} = \frac{y!}{(y-x)!}$$

組合せ $\boxed{g} \boxed{Cy,x}$ キーを押すと、 y 個の異なるものから x 個取出す(順序に関係なく)方法(つまり組合せ)を計算します。(x を変えて計算するときは y も入れ直す必要があります。) 計算式は次の通りです。

$$C_{y,x} = \frac{y!}{x!(y-x)!}$$

順列と組合せを計算するときは次のようにします。

1. 種類の数(y)をキーインします。
2. \boxed{ENTER} を押します。
3. 個数(x)をキーインします。
4. $\boxed{f} \boxed{Py,x}$ または $\boxed{g} \boxed{Cy,x}$ を押します。



答の p や c が Xレジスタに入って表示され、Z や Tレジスタの内容がスタック下降し、個数 x はラスト x レジスタに入っています。

順列や組合せの計算用の数は必ず正の整数(0を含む)だけです。

注 $\boxed{Py,x}$ や $\boxed{Cy,x}$ の計算はキーインした y と x の値が大きいほど答が出るまでに時間がかかります。 $\boxed{Py,x}$ と $\boxed{Cy,x}$ の計算に使える最大の数は $10^{15}-1$ です。 $\boxed{Py,x}$ と $\boxed{Cy,x}$ の計算中は running の文字が点滅します。

計算目的	キー操作例	表示
順列 10個から3個取っ たときの並べ方	10 ENTER 3 f Py,x	10.0000 720.0000
組合せ 10個から3個取 出す方法	10 ENTER 3 g Cy,x	10.0000 120.0000

統計用関数

乱数 (random number) 発生

HP-11Cは乱数の種(核ともいいます)をあなたがキーインするか、計算機中に0を自動的に入れさせてから乱数キーを押すと $0 \leq r < 1$ の疑似一様乱数を計算します。不揮発性メモリーなので別の乱数の種をストアするまでは、それまでの乱数列(一連の乱数を並べると数列になるから)の続きを計算します。この乱数発生方法はDonald E. Knuthの乱数列の検定に合格しています。

乱数の種 新しい乱数列を作りたいときは、 $0 \leq n < 1$ の任意の数*をキーインし**STO** **f** **RAN#**と押ししてストアします。乱数を作るには**f** **RAN#**と押します。すると新しい乱数がXレジスタに入って表示します。(**f** **RAN#**を押すとスタック内の数値が移動します。)新しい乱数を計算するとそれを次の乱数の種としてストアするという順序を繰り返します。不揮発性メモリーをリセットしたり、電池が切れると乱数の種は0になってしまいます。(乱数の種を同じにすると乱数列も同じになります。)

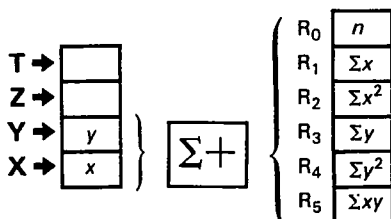
*この数が1以上のときは、この数を乱数の種に使えるようにするために、一番左の有効数字の一つ左まで小数点が自動的に移動します。例えば123や12.3あるいは1.23を**STO** **f** **RAN#**でストアすると、どれも0.123をストアしたのと同じになります。

乱数発生	キー操作列	表示
0.5764を乱数の種 (任意の数)として キーインする 種をストア	$\boxed{0.5764}$	0.5764
	$\boxed{\text{STO}} \boxed{f}$ $\boxed{\text{RAN\#}}$	0.5764
この種を使って乱 数列を求める	$\boxed{f} \boxed{\text{RAN\#}}$	0.3422
	$\boxed{f} \boxed{\text{RAN\#}}$	0.2809
	⋮	⋮

この乱数発生についてはこの本の第11章の乱数 (217ページ) もご覧ください。

統計用データの集計

$\boxed{\Sigma+}$ は X と Y レジスタの統計用データを集計するための二項演算キーです。計算結果は自動的に $R_0 \sim R_5$ に集計します。

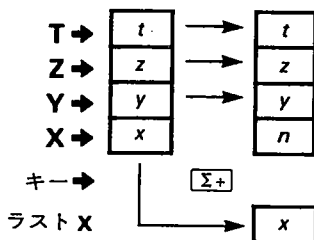


集計を始める前に $R_0 \sim R_5$ に別の計算結果が入っていると、正しい集計結果が求まりません。そこで集計を始める前に \boxed{f} CLEAR $\boxed{\Sigma}$ ($R_0 \sim R_5$ とスタックをクリアする) を押して集計用レジスタを全部 0 にする必要があります。

$\Sigma+$ キーを押すと下表のような集計をします。

レジスタ	内 容	
R ₀	n	集計したデータの個数(組数),つまり $\Sigma+$ を押した回数(n がXレジスタに入って表示される)
R ₁	Σx	x の累計
R ₂	Σx^2	x の二乗の累計
R ₃	Σy	y の累計
R ₄	Σy^2	y の二乗の累計
R ₅	Σxy	x と y の積の累計

$\Sigma+$ キーを押すとそれまでXレジスタにあった数値はラストXレジスタに移り,その代りに n がXレジスタに入ります。Yレジスタにあった数値はそのまま残ります。



新しい数値を1個キーインすると,それまでXレジスタにあった n が消えるだけでスタック上昇は生じません。

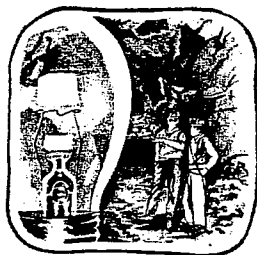
集計したい数値が x と y でセットになったものでなく, x だけのときは $\Sigma+$ を押す前にYレジスタを0にする必要があります。(集計する前に f CLEAR Σ を押せば, Σ レジスタのR₀~R₅だけでなくYレジスタまでクリアするので,安心して x だけの集計もできます。)

x や y の集計のときに、場合によっては x どうしや y どうしの差が x や y に比べて非常に小さいときがあると思います。このようなときには x や y のおよその平均を差引いてそれを集計してください。そして \bar{x} や \bar{y} を計算したり L.R. の y 切片を計算してから、差引いたおよその平均をたしてください。例えば x が 665999, 600000, 600001 とするとそれぞれから 600000 を引いた $-1, 0, 1$ を集計します。 x を計算してから 600000 を加えれば正しい答になります。この方法をデータの正規化といいます。 s, r や L.R. や \hat{y} が計算できなかつたり、これらのキーを押したときに Error 3 の表示がでたら、上記のデータの正規化をして集計しなおしてください。

注 $\Sigma+$ や $\Sigma-$ 操作中に $R_0 \sim R_5$ のレジスタでオーバーフローが発生しても Error 1 の表示が出ません。

データの集計途中や集計後に集計用レジスタの累計を見たいときは、 RCL を押してから目的のレジスタ番号を押すと X レジスタにその数値が入って表示します。 Σx と Σy を一度に呼出すには RCL $\Sigma+$ と押します。こうすると R_1 の Σx が X レジスタに入り、 R_3 の Σy が Y レジスタに入ります。(RCL $\Sigma+$ を押すと RCL 3, RCL 1 と押したのと同じでスタック上昇を生じます。)

例 電気エネルギーを研究している Helen I. Voltz さんは世界全体の1972～1976年の石炭採掘量と発電量の関係を調べたいと考えました。HP-11Cで石炭採掘量と発電量を集計して Σx , Σx^2 , Σy , Σy^2 , Σxy を計算してください。



年	1972	1973	1974	1975	1976
石炭採掘量 (y) (単位: 10 億トン)	1.761	5.963	1.792	1.884	1.943
発電量 (x) (単位: 10 億メガワット時)	5.552	5.963	6.135	6.313	6.713

キー操作 表示

\boxed{f} CLEAR $\boxed{\Sigma}$	0.0000	集計用レジスタ ($R_0 \sim R_5$ とスタック) のクリア
1.761 \boxed{ENTER}	1.7610	
5.552 $\boxed{\Sigma+}$	1.0000	1972年のデータ
1.775 \boxed{ENTER}	1.7750	
5.963 $\boxed{\Sigma+}$	2.0000	1973年のデータ
1.792 \boxed{ENTER}	1.7920	
6.135 $\boxed{\Sigma+}$	3.0000	1974年のデータ
1.884 \boxed{ENTER}	1.8840	
6.313 $\boxed{\Sigma+}$	4.0000	1975年のデータ
1.943 \boxed{ENTER}	1.9430	
6.713 $\boxed{\Sigma+}$	5.0000	1976年のデータ
\boxed{RCL} 1	30.6760	R_1 の x の累計値 (Σx)
\boxed{RCL} 2	188.9386	R_2 の x^2 の累計値 (Σx^2)
\boxed{RCL} 3	9.1550	R_3 の y の累計値 (Σy)
\boxed{RCL} 4	16.7877	R_4 の y^2 の累計値 (Σy^2)
\boxed{RCL} 5	56.2924	R_5 の x と y の積の累計値 (Σxy)

集計データの訂正

データ集計中にデータを間違えて集計してしまっても、集計データを簡単に訂正できます。

1. XとYレジスタに間違えたデータを入れる。
2. Σ^- を押して間違えたデータを削除する。
3. 正しい x と y をキーインする。 x と y のセットのうち片方だけを間違えても、両方を削除して改めて正しいデータをキーインする必要があります。
4. Σ^+ キーを押す。

注 Σ^- を押して間違えた x と y のデータを削除しても、有効数字の10桁目に四捨五入の誤差が発生したために誤データ集計前と誤データ集計・削除後の数値がぴったり一致しないことがあります。このようなことはデータの桁数が6桁以上でなければ発生しませんが、このような心配があったら統計レジスタをクリアして集計データをもう一度（間違えないように）入れなおしてください。

例 Voltzさんが先程のデータを全部入れた後で1976年のデータが1.943でなくて1.946だったのに気がつきました。 Σ^- を使って間違えたデータを削除してから正しいデータを入れなおしてください。

キー操作	表示	
1.943 Σ^-	1.9430	間違えたデータを削除すると、入れたデータの組数が一つ減るので4になる。
6.713 Σ^-	4.0000	
1.946 Σ^+	1.9460	正しい数値を入れ直した。データの組数が元の5に戻った。
6.713 Σ^+	5.0000	

この集計データはこれから各種の計算に使いますのでクリアしないでください。

平均

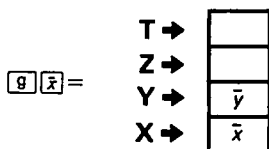
\bar{x} キーを使えば、集計用レジスタの R₁ と R₃ をわざわざ使わなくても x と y の平均を計算できます。 $\text{g} \bar{x}$ を押すと：

1. 57ページの $\text{RCL} \Sigma+$ と同じようにスタック上昇を生じます。
2. x の平均 (\bar{x}) は集計用レジスタに集計した R₁ (Σx) を R₀ (n) で割って計算します。 y の平均 (\bar{y}) は集計用レジスタに集計した R₃ (Σy) を R₀ (n) で割って計算します。計算式は次のとおりです。

$$\bar{x} = \frac{\Sigma x}{n}$$

$$\bar{y} = \frac{\Sigma y}{n}$$

3. 計算した \bar{x} と \bar{y} がそれぞれ X と Y レジスタに入ります。



例 5年分の(訂正後の)集計データを利用して石炭採掘量と発電量の平均を計算してください。

キー操作

 $\text{g} \bar{x}$

表示

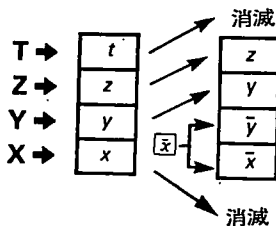
6.1352

5年間の発電量 (Xレジスタに入れたデータ) の平均

 $\text{x} \geq y$ **1.8316**

5年間の石炭採掘量 (Yレジスタに入れたデータ) の平均

下図は \boxed{x} キーを押したときのスタック内のデータ移動の説明図です。



前の統計データは次の例題にも使いますのでクリヤしないでください。

標準偏差

$\boxed{9} \boxed{s}$ キーを押すと集計したデータの標準偏差(データの平均からの離れ具合)を計算します。HP-11C は下式を使って s_x (集計した x の標準偏差) と s_y (集計した y の標準偏差) を同時に計算します。

$$s_x = \sqrt{\frac{n\sum x^2 - (\sum x)^2}{n(n-1)}} \quad s_y = \sqrt{\frac{n\sum y^2 - (\sum y)^2}{n(n-1)}}$$

計算式の答はあるデータ全体(母集団)から抜取ったデータ(サンプリング・データ)だけを使ってデータ全体の標準偏差を推定するために使うサンプルの標準偏差です。 $\boxed{9} \boxed{s}$ を押すと次のようになります。

- 57 ページの $\boxed{RCL} \boxed{\Sigma+}$ のときと同じようにスタック上昇をします。
- 集計用レジスタの $R_2(\Sigma x^2)$, $R_1(\Sigma x)$ と $R_0(n)$ のデータを使って上式のとおり x の標準偏差 (s_x) を計算して X レジスタに入ります。

3. 集計用レジスタの $R_4(\Sigma y^2)$, $R_3(\Sigma y)$ と $R_0(n)$ のデータを使って上式のとおり y の標準偏差 (s_y) を計算して Y レジスタに入ります。

例 5年分の(訂正後の)集計データを利用して石炭採掘量と発電量の標準偏差を計算してください。

キー操作

$\boxed{g} \boxed{s}$

表示

0.4287

5年間の発電量(Xレジスタに入れたデータ)の標準偏差

$\boxed{x} \boxed{\geq} \boxed{y}$

0.0800

5年間の石炭採掘量(Yレジスタに入れたデータ)の標準偏差

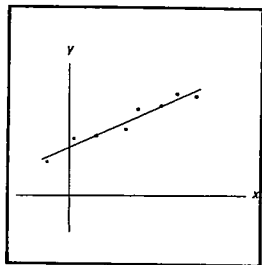
この集計データは次の例にも使いますのでクリアしないでください。

入力したデータが抜取りデータでなくて、母集団の全データのときには母集団の標準偏差が必要なことがあります。サンプルの標準偏差に $\sqrt{\frac{n-1}{n}}$ を掛ければ母集団の標準偏差になります。入力データの値が小さかったり、データ数が多ければこの誤差は無視できます。しかし母集団の標準偏差を正しく求めたいときは \boxed{s} を求めてから上の係数を掛けるか、次の簡便法を使います。

それは $\boxed{g} \boxed{\bar{x}}$ を押して平均を求め $\boxed{\Sigma+}$ でその平均を集計してから $\boxed{g} \boxed{s}$ を押します。これが母集団の標準偏差になります。ただしこの後で L.R., r や \hat{y} の計算をするには $\boxed{g} \boxed{\bar{x}}$ を押してから $\boxed{g} \boxed{\Sigma-}$ を押して前の状態に戻します。

直線のあてはめ (直線回帰)

直線のあてはめは2組以上のデータにあてはまる統計的に最適な直線の計算式を求めることです。xとyの組になったデータを2組以上R0~R5に集計してから、 $\boxed{f} \boxed{L.R.}$ を押すと最小二乗法で最適な直線 $y = Ax + B$ の係数 A, B を計算します。



HP-11Cでこの直線のあてはめをするには、 $\boxed{\Sigma+}$ キーを使って2組以上のxとyを集計します。それからを押します。すると：

1. 57 ページの $\boxed{RCL} \boxed{\Sigma+}$ のようにスタック上昇を生じます。
2. 直線の傾斜 (勾配) A と、y 軸の切片 B を次式のとおり最小二乗法で計算します。

$$A = \frac{n \Sigma xy - \Sigma x \Sigma y}{n \Sigma x^2 - (\Sigma x)^2}$$

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

3. 傾斜AはYレジスタに、切片BはXレジスタに入ります。

スタック内のデータは右図のようになります。

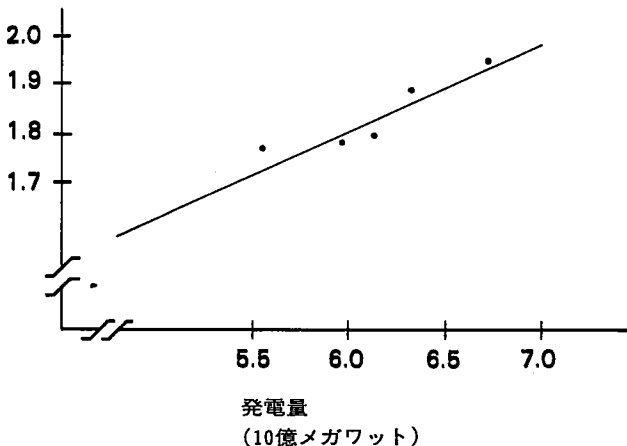
例 先程の石炭と発電データにあてはまる直線の傾斜と切片を求めてください。

$\boxed{f} \boxed{L.R.} =$

T →	
Z →	
Y →	勾配 (A)
X →	y 軸の切片 (B)

方法 石炭採掘量と発電量をグラフに書くと次図のようになります。HP-11Cにはこのデータが $\boxed{\Sigma+}$ を使って入れてあるので $\boxed{f} \boxed{L.R.}$ を押すだけです。

石炭採掘量
(10億トン)



キー操作

\boxed{f} $\boxed{L.R.}$

$\boxed{x \cong y}$

表示

0.7773

0.1718

y 軸の切片 (y 軸との交点)

直線の傾斜 (勾配)

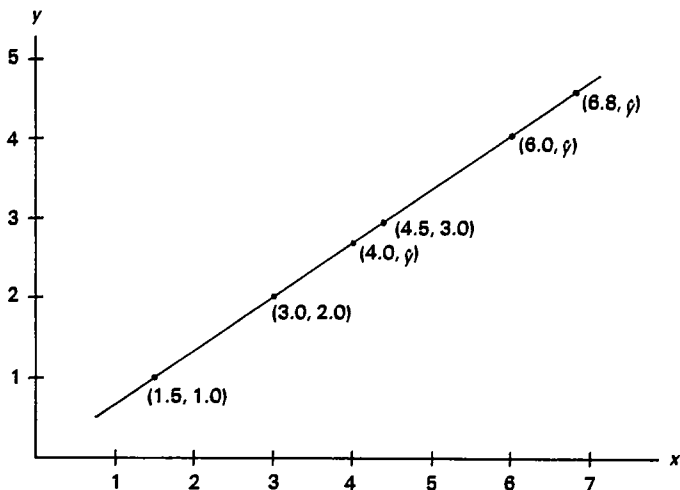
集計してあるデータは次の例にも使うのでクリアしないでください。

y の推定と相関係数

$\boxed{\hat{y}, r}$ を使うと y の推定値 (\hat{y}) が X レジスタに、相関係数 r が Y レジスタに入ります。

y の推定 x のときの y の推定値 (\hat{y}) を求めるには、まず x を入力して $\boxed{\hat{y}, r}$ を押すと、集計用レジスタ R0~R5 の値を使って \hat{y} を計算します。

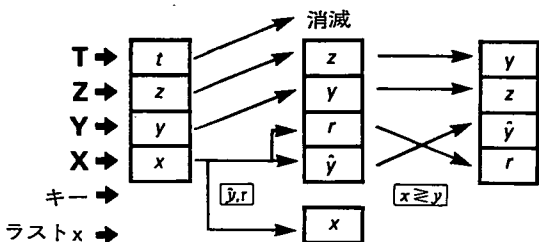
$$y = \frac{\sum y \{ n \sum x^2 - (\sum x)^2 \} + (n x - \sum x) (n \sum xy - \sum x \sum y)}{n \{ n \sum x^2 - (\sum x)^2 \}}$$



相関係数 直線のあてはめと y の推定は x と y のデータがある程度の直線関係（つまり前記の一次式があてはまる）があるものとして計算しています。相関係数は入力したデータが直線にどの程度一致しているかを表わすものです。相関係数 r は $-1 \leq r \leq +1$ の範囲です。 $r = +1$ のときは直線がデータにぴったり一致していて右上り（勾配がプラス）です。 $r = -1$ のときは直線がデータにぴったり一致していて右下り（勾配がマイナス）です。 $r = 0$ のときはデータに直線がうまくあてはまらないことを表わします。 \boxed{f} $\boxed{y,r}$ を押すと集計用レジスタ $R_0 \sim R_6$ を使って相関係数の計算をします。表示している Xレジスタの値は前例のように前もって x を入れたとき以外は意味のない値です。次に $\boxed{x \rightleftharpoons y}$ を押すと X と Y レジスタの内容が入替わって、相関係数 r を表示します。

$$r = \frac{\frac{1}{n-1}(\sum xy - \frac{\sum x \sum y}{n})}{\sqrt{\frac{1}{n-1} \left\{ \sum x^2 - \frac{(\sum x)^2}{n} \right\}} \sqrt{\frac{1}{n-1} \left\{ \sum y^2 - \frac{(\sum y)^2}{n} \right\}}}$$

第2章で説明したラスト x は $\boxed{\bar{x}}$, \boxed{s} と $\boxed{L.R.}$ のときにはその直前の x の値はラスト x レジスタには入りません。しかし $\boxed{f} \boxed{\hat{y}.r}$ のときにはその前の x の値を使って \hat{y} の計算をするので、 x の値はラスト x レジスタに入ります。そしてどの場合も同時にスタック上昇が起ります。



例 前の統計データを使って1977年の石炭の必要量 (\hat{y}) を計算したいと思います。それには1977年の予想発電量 (x) をキーインして $\boxed{f} \boxed{\hat{y}.r}$ を押します。このキーを押すと自動的に相関係数も計算できるので、 \hat{y} を表示したあとで $\boxed{x \geq y}$ を押せば Voltz さんのデータと直線とのばらつき具合もすぐ見られます。

キー操作

表示

7.1417

7.1417

Voltzさんの予想した1977年の発電量。

 $\boxed{f} \boxed{\hat{y}.r}$

2.0046

これに相当する石炭の採炭予想量。

 $\boxed{x \geq y}$

0.9211

データが直線にかなり接近している。

第4章 表示指定

HP-11Cのスイッチを入れると、不揮発性メモリーにスイッチを切る前の情報が入っているため、スイッチを切る前の状態に戻ります。表示桁数も同様ですが、HP-11C内部では仮数部（有効数字）10桁と10の指数部2桁の容量があります。例えば小数点以下4桁まで表示するようにしてあるときに π を表示させると3.1416を表示しますが、内部では $3.141592654 \times 10^{00}$ になっています。

$$3.141592654 \times 10^{00}$$

この部分だけを表示する（正しくは小数点以下4桁に四捨五入した（小数点以下5桁目で四捨五入した）結果を表示）

内部にはこの部分の数値も入っている

表示形式の指定

HP-11Cは **[FIX]**、**[SCI]** と **[ENG]** の3種の表示形式があり、この後に0～9の桁数を指定します。

下図は123.456を3種の表示形式で4桁指定したときの表示例です。

f	[FIX]	4 :	123.456.0000
f	[SCI]	4 :	1.2346 05
f	[ENG]	4 :	123.46 03

固定小数点表示

[FIX]（fixed decimal）は数値を固定小数点表示に指定するときに使います。一たん **[FIX]** に指定すれば、それでは表示できない大きな数や小さな数を **[SCI]** で表示する以外は全部固定小数点表示になります。**[FIX]** で表示できない大きな数や小さな数も、計算を続けて **[FIX]** で表示できる数値になれば自動的に指定した **[FIX]** 表示に戻ります。

f SCI 4	1.2346 02	四捨五入して 1.2346×10^2 (小数点以下 5 桁目で四捨五入)
f SCI 6	1.234568 02	四捨五入して 1.234568×10^2 (小数点以下 7 桁目で四捨五入)

上の例のように **SCI** で指定した表示末尾の桁は四捨五入した結果によっては数字が変わることがあります。桁指定を 7 以上にしても小数点以下の表示桁数は 6 桁よりも増えませんが、内部では指定桁数通りに四捨五入しています。上の例のままですら次の操作をやっても表示桁数はそのままですが、表示してない部分で指定通りに表示のための四捨五入をしています。*

キー操作

表示

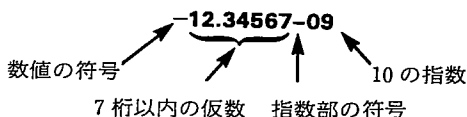
f SCI 7	1.234567 02	四捨五入して小数点以下 7 桁にしているが、 SCI では小数点以下 7 桁目が表示できないので、表示では四捨五入してないように見える。
f SCI 8	1.234567 02	四捨五入して小数点以下 8 桁にしているが表示は変化しない。
f SCI 9	1.234567 02	四捨五入して小数点以下 9 桁にしているが表示は変化しない。

* 数値の末尾近くの 9 の後に数字が続くと、**SCI** 7 や 8 を指定するとその結果が誇張されて表示されることがあります。例えば 1.00000094 では表示に変化はありませんが、1.00000095 (…96 から…99 までも同じ) のときに **SCI** 6 では表示末尾が 1 になるのは当然ですが **SCI** 7 のときも同じ表示になります。

工学浮動小数点表示

ENG (engineering) は工学浮動小数点表示で、次の点を除いては **SCI** と同じです。

- **ENG** では指数部が必ず3の整数倍になります。
- 最初の1桁目の数字の右側の部分の桁数が指定通りの桁数になります。



工学浮動小数点表示では必ず始めの1桁は表示します。それより右側の桁数は **f** **ENG** で指定した桁数になるように四捨五入した結果を表示します。例えば

キー操作	表 示	
f ENG 1	0.012345 12. -03	工学浮動小数点表示では最初の1桁目の右側の桁数が指定した通りの桁数になるように四捨五入する。10の指数は必ず3の倍数になるように小数点が移動する。
f ENG 3	12.35 -03	全部で4桁表示になるように四捨五人。
f ENG 6	12.34500-03	
f ENG 0	10. -03	有効数字が1桁になるように四捨五入。

ENG では10の指数が必ず3の倍数になるように小数点の位置が移動するので気をつけてください。

キー操作	表示		
\boxed{f} \boxed{ENG} 2	12.3	-03	前の数値を \boxed{ENG} 2 表示にした。
10 $\boxed{\times}$	123.	-03	小数点の位置だけが動いた。

大きな数値のキーイン

\boxed{EEX} (enter exponent) は 10 の指数部のキーイン用に使います。 \boxed{EEX} を押す前に仮数をキーインし、それから \boxed{EEX} を押して最後に指数をキーインします。例えば 95,600 をアボガドロ数 (6.0222×10^{26} kmol⁻¹) で割ってみましょう。

キー操作	表示		
\boxed{f} \boxed{FIX} 4			表示を \boxed{FIX} 4 に戻す。
95600 \boxed{ENTER}	95,600.000		
6.0222	6.0222		
\boxed{EEX}	6.0222	00	00 はこれから指数を入れる位置。
26	6.0222	26	(6.0222×10^{26})
$\boxed{+}$	1.5875	-22	kmol。

10 の指数が負数のときには、まず仮数をキーインし \boxed{EEX} を押し、それから指数の符号をマイナスにするために \boxed{CHS} (change sign) を押した後で指数をキーインします。例えばプランク (Planck) の常数 (6.6262×10^{-34} ジュール・秒) をキーインし、それを 50 倍してみましょう。

キー操作	表示		
6.6262 \boxed{EEX}	6.6262	00	
\boxed{CHS}	6.6262	-00	
3	6.6262	-03	
4	6.6262	-34	
\boxed{ENTER}	6.6262	-34	
50 $\boxed{\times}$	3.3131	-32	ジュール・秒

注 \boxed{EEX} を押した後で指数をキーインしなくても、計算機内部ではその情報がそのまま残っています。

72 第4章 表示指定

整数部の桁数が8桁以上になったときや、小数点以下で最初の有効数字までに0が6桁以上になったときには **[EEX]** キーを押さないでください。このような数のときには指数を大きくしたり小さくして調節してください。例えば $123456789.8 \times 10^{23}$ のときには $1234567.898 \times 10^{25}$ のようにキーインし、 $0.00000025 \times 10^{-15}$ のときは 2.5×10^{-22} としてキーインしてください。

仮数の表示 計算のスタックや数値レジスタ内の数値は全部10桁の仮数と2桁の指数として記憶しています。この10桁の仮数全体を見たいときにはまずXレジスタにその数値を移してから **[f] CLEAR [PREFIX]** を押してこのキーを押したままにします。するとキーを放すまで仮数の全桁を見ることができ、キーを放してもしばらくの間表示し続けます。

キー操作

表示

[f] π

3.1416

[f] CLEAR [PREFIX]

(キーを押したまま)

3141592654

10桁目の数字

前にも説明したようにHP-11Cはどんな数値でもそのときの表示形式に無関係に仮数(有効数字)を10桁記憶しています。一つ一つの計算や連続計算の結果は必ず10桁になるように四捨五入しています。例えば π や $2/3$ は無限小数です(3.1415926535...や0.6666666666...となります)。HP-11Cでは有効数字が10桁に限られるため、11桁目で四捨五入してしまうので10桁目に誤差が出ることがあります。この誤差は長い計算で計算回数が多くなるほど大きくなりますが、特殊の計算以外は気にすることはないでしょう。数値解析などのときの切上げ切捨て誤差についてはこの本の範囲を超えていますので、誤差論などの本をご覧ください。

第 2 部
HP-11C の
プログラム

第5章 プログラム作成の基礎知識

基礎知識の要点

プログラムとは？

計算機に覚えこませたキーの操作順序をプログラム命令といい、全体をプログラムと言います。一つのプログラム命令はキーを1～3個押す操作に相当します。プログラムを走らせると、一つ一つのプログラム命令ごとに手でキーを操作したときとまったく同じようにスタック内のデータが変化します。プログラムの終わりまで来ると、手でキー操作したときのように答だけを表示します。HP-11Cのプログラム作成は誰にでもできます。

なぜプログラムを書く必要があるのでしょうか？

プログラムがあれば計算時間が短縮できます。一たんプログラムを作成して計算機中に記録してしまえば、実際の計算のときにいちいちキー操作に気を使う必要がなくなります。つまり計算機に計算させることができるわけです。そしてプログラム内容が簡単にチェックできるので、計算した答がおかしかったらプログラムを記憶させたときのキー操作が悪かったのかどうかすぐ調べられます。

これからの説明は HP-11C のプログラム作成や操作のことが主になっています。プログラムを作るときの考えかたなどは、この本の第3部のプログラム作成の技術(206ページ)をお読みください。

プログラム操作

メモリーの自動配分

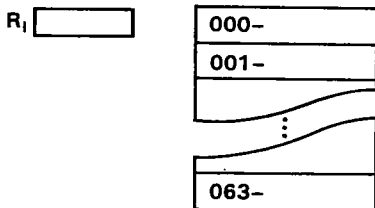
HP-11Cのメモリーのプログラム用と数値記憶用との配分は計算

機が自動的に調整しています。内部で自動的に配分しているので表示やキー操作には変化がありません。それでこれだけ知っておいてください。

1. メモリー配分の原因
2. メモリー配分の結果がどうなるか

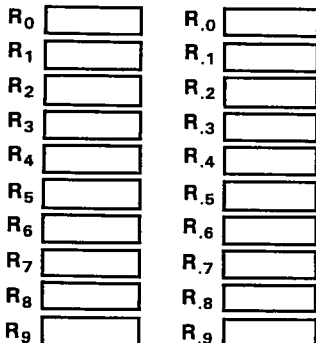
プログラム・メモリーをクリアしたり、不揮発性メモリーをリセットすると計算機のメモリーは数値記憶用が20個とIレジスタ、プログラム用が63ラインになります。

HP-11Cのメモリー配分
専用のメモリー

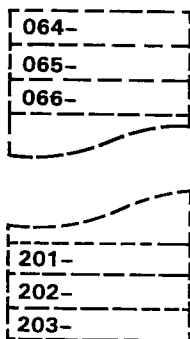


共用（両用）のメモリー（初めの配分）

数値記憶レジスタ20個



プログラム用は0



プログラムとしてキーインした命令はプログラム・メモリー中に順番に記録されます。63ラインまでは最初からあるプログラム・メモリーに入りますが、64ライン目の命令をキーインすると記憶レジスタの R_9 が自動的に7ライン分のプログラム・メモリーに変化してそこに記録されます(まだ6ライン分の余裕があります)。プログラムが70ラインまでなら記憶レジスタは20個使えます。次に71ライン目の命令をキーインすると直ちに記憶レジスタの R_8 が7ライン分のプログラム・メモリーに変化します。このようにして記憶レジスタ20個分 (R_9 から R_9 , R_9 から R_0) がプログラム用に配分されます。

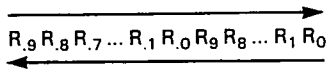
記憶レジスタ20個がプログラム用になるとプログラム203ライン分になり、記憶レジスタは1個 (Iレジスター R_1) だけになります。下表はプログラムの何ライン目がどのレジスタを使うかを表にまとめたものです。

R_9	064—070	R_9	134—140
R_8	071—077	R_8	141—147
R_7	078—084	R_7	148—154
R_6	085—091	R_6	155—161
R_5	092—098	R_5	162—168
R_4	099—105	R_4	169—175
R_3	106—112	R_3	176—182
R_2	113—119	R_2	183—189
R_1	120—126	R_1	190—196
R_0	127—133	R_0	197—203

記憶レジスタとプログラム・メモリーの配分

プログラム・メモリー中の命令を削除するとそれに応じてプログラム・メモリーが先程と逆の順序で順番に記憶レジスタに戻りません。

記憶レジスタからプログラム・メモリーに変化する順番

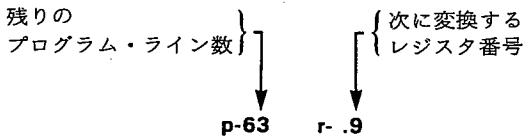


プログラム・メモリーから記憶レジスタに戻る順番

もっと詳しい説明は付録Cメモリーの自動配分にまとめてあります。

MEM

プログラム・モード（計算機にプログラムを記憶させる状態、以下 PRGM モードと略記）でもそれ以外でも、そのときのメモリー配分を見るには **MEM** (memory) キーを押したままにすると、(1)次の数値メモリーをプログラム用メモリーに変換しない状態でプログラム可能な残りライン数、(2)次にプログラム用メモリーに変換するレジスタ番号を表示します。（**MEM** 操作の説明は付録Cメモリーの自動配分（227ページ）に記載しました。）



（これはプログラム・メモリーをクリアしたときの **MEM** の表示です）

キーコードとライン番号

計算機を PRGM モードにしてプログラム命令を一つキーインすると、プログラムのライン番号とその命令のキーコードを表示します。一つのプログラム命令は1～3個のキーコードですが、これは命令を入れたときのキーを押した回数に相当します。各キーコードは数字キー以外は2桁の数値で、左側がそのキーの上からの段数、右側が左からの列数のマトリクス表示です。数字キーは

その数字だけの1桁表示です。

017	42, 21, 11			
ライン番号	↓			
4	2	1	そのキーの上からの段数	
2	1	1	そのキーの左からの列数	

キー操作の省略

PRGM モードや計算のときに一連のキー操作の途中で **f** キーを押すのを省略できる操作があります。(**f** キーを押さなくてもよいキー操作をプログラム命令として入れると **f** キーのキーコードは表示しません。) 例えば **STO** **RAN#** とキーを押しても **STO** **f** **RAN#** とキーを押しても同じ結果になります。このように **f** キー操作を省略できるものはその操作の説明のときに明記いたします。

プログラム操作機能

プログラムと計算の切替え **g** **P/R** (program/run) キーを押すと計算機が PRGM モードと計算 (run) モードに切替ります。計算機が PRGM モードに切替わると表示窓に PRGM の文字を表示して、プログラム命令の入力と削除ができるようになります。計算モードではプログラム・メモリーに入れたプログラムを走らせることも、手操作だけで計算することもできます。

プログラム・メモリーのクリア PRGM モードで **f** CLEAR **PRGM** キーを押すと、不揮発性メモリー中のプログラム命令を全部削除(クリア)すると同時にメモリーを数値メモリーを 21 個、プログラム用メモリーを 63 ライン分に配分します。計算モードで **f** CLEAR **PRGM** を押すと計算機をライン 000 にセットするだけで、プログラム命令をクリアしません。

ライン 000 にセット PRGM モードでも計算モードでも **GTO** (go to) **000** と押すと計算機をライン 000 にセットできます。

ラベル HP-11Cのラベルはプログラムそのものやプログラムの分岐先やサブルーチンの目印(番地のようなもの)です。HP-11Cではラベル符号(A～E)とラベル番号(0～9)があって、PRGMモードで[f] [LBL] (label) キーを押してから符号や数字のキーを押してプログラム・メモリーに記憶させます。計算モードで[f] キーを押してからラベル符号のキーを押すと、その符号の部分からプログラム計算を開始します。プログラム中のラベル番号(0～9)は普通そのプログラム中の一部分の目印(分岐先やサブルーチンなど)に使います。[GSB] を押してその数字キーを押せばそのラベル番号から計算がスタートします。

リターン [RTN] (return) 命令はプログラムの最後に入れ、プログラム計算中にRTNに出会うとライン000に戻って計算が終了します。

[R/S] プログラム計算中に[R/S] (run/stop) 命令に出会うか[R/S] キーを押すとプログラム計算を停止します。プログラムが停止しているときに[R/S] を押すと、計算機はそのときのプログラム位置から計算を開始します。

ポーズ プログラム計算中に[f] [PSE] (pause) 命令に出会うと、一時的に(約1秒間)プログラム走行を停止して、そのときのXレジスタの内容を表示し、約1秒後に再スタートします。

ユーザー・モード

ユーザー・モード(以下 USER モードと略します)はプログラム計算を始めるときにキー操作を簡単にする機能です。USER モードにすると上から一段目のキーを押す前の[f] キーの機能が逆になります。

[f] [USER]

0.0000
USER

USER モード表示

USER モードにすると

[f]の後でキーを押すと	→	A	B	C	D	E
そのままキーを押すと	→	\sqrt{x}	e^x	10^x	y^x	$1/x$
[g]の後でキーを押すと	→	x^2	LN	LOG	%	$\Delta\%$

計算モードのときに USER モードにして **f** を押さずにこの **A** ~ **E** のキーを押すと、プログラム中のこのラベル符号の部分から計算がスタートします。

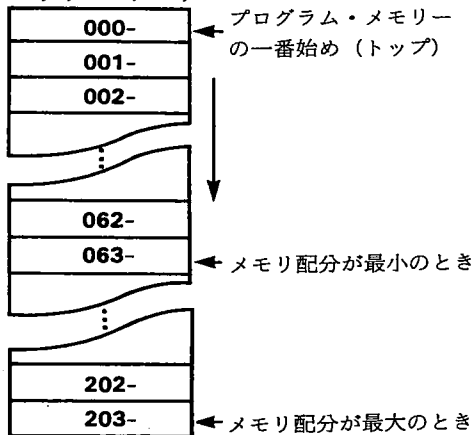
注： USER モードにする必要があるとき以外は、プログラムのスタート位置の混乱を防ぐために USER モードにしないでください。

プログラム・メモリー

この本の始めの部分で熱損失の計算をしたときに、手計算をしたときのキー操作順序どおりにプログラムとして書込みました。このキー操作順序をプログラム・メモリーに記憶させたわけです。

GTO **000** と押すとプログラム・メモリーの一番始め(トップ)に戻ります。まだこのキー操作をしていなかったら PRGM モードに切替えるために **g** **P/R** と押してください。(PRGM モードにすると表示に PRGM の文字が見えます。) **000-** を表示していたらプログラム・メモリーのトップに戻っています。

プログラム・メモリー



プログラム・メモリーはスタック、LAST x, R_i レジスタやまだプログラム・メモリーに変換していない数値メモリーとは別のメモリーです。

PRGM モードにしたときに左端に表示している3桁の数値は、計算機がそのときにプログラム・メモリーのどこにいるかを示すライン番号です。[f] CLEAR [PRGM] を押してから、12ページの熱損失プログラムの1ライン目の[f] [LBL] [A]を押すと下のように表示が変わります。

001-42,21,11
 ライン番号 ← ← キーコード

表示左端の数値でわかるように、計算機はプログラム・メモリーの001ラインにセットされました。その右側の数値はプログラム・メモリーに記憶させたキー操作のキーコードです。3を押すと次のようになります。

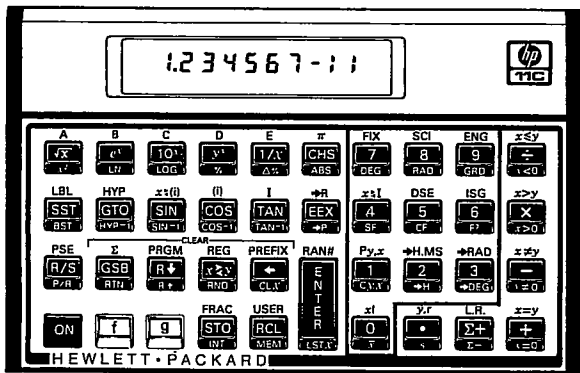
002- 3
 ライン番号 ← ← キーコード

左端の002はプログラムの2ライン目だという表示です。

プログラム・ラインの各ラインごとに一つのプログラム命令（キー操作1～3回分）を記憶します。例えば [CHS] のようにキー操作が1回だけの命令も、[STO] [+] 6（そのとき表示しているXレジスタの数値をR₆に加算する）のようにキー操作が3回の命令も1ラインに納まります。プログラム命令として入れた（記憶させた）キー操作はキーコードとして表示します。

キーコードの解読

77ページに記載したようにHP-11Cのキー位置は上からの段数と左からの列数のマトリクス表示を採用しています。キーの段数は1～4で、列数は1～10です。（HP-11Cの10列目のキーコードは0を使っているため、"20"は上から2段目左から10列目の[x]キーのことで）ただし0～9の数字キーは見やすくするためにそのキーの上面の数字どおりの1桁の数値になります。



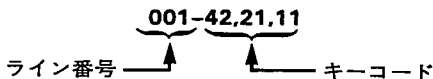
1桁のキー・コード

1ラインに一つから三つのキーコードを表示した例は次のとおりです。



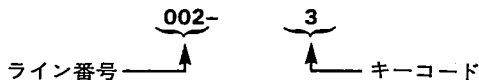
上から 2 上から 4 3 上から 4 4 数字の“5”
 左から 3 左から 3 3 左から 4 10

それでは先ほど (81 ページで) キーインしたプログラム命令を見てみましょう。 を押してください。熱損失のプログラムの 1 ライン目が見えるはずですよ。



左端の 001- はプログラム・メモリーの 1 ライン目です。次の 2 桁の 42 は (上から 4, 左から 2) のキー、21 は (上から 2, 右

から1), 11は **A** (上から1, 左から1) のキーのことです。このように数字キーを使う関数キー以外はこのように全部2桁のキーコードになります。**SST** を1回押して次を見ましょう。HP-11Cは熱損失のプログラムの2番目の命令のキーコードを表示しているはずですが。



002- はプログラムのライン番号, この3は数字キーの3です。右の枠の内は数字キーの3で指定できる命令のキーコードです。

キー	キーコード
f → RAD	→ 42 3
3	→ 3
g → DEG	→ 43 3

熱損失の残りのプログラム命令のキーコードは次のとおりです。残りの命令をキーインしてキーコード表示を確認してください。

キー操作	表示	
x	003- 20	上から2段目, 左から10列目のキー。
1	004- 1	数字キーの1。
.	005- 48	上から4段目, 左から8列目のキー。
7	006- 7	数字キーの7。
8	007- 8	数字キーの8。
x	008- 20	上から2段目, 左から10列目のキー。
g RTN	009- 43 32	プログラムの終り。
g P/R		計算機を計算モードに戻す。

プログラム作成の手順

これまで説明してきたことは一つ一つのプログラム命令に関することでした。これからプログラムを一つ作成する手順を順番に説明したいと思います。

$A = \pi r^2$ の計算式で円の面積を手計算するときにはまず r をキーイン

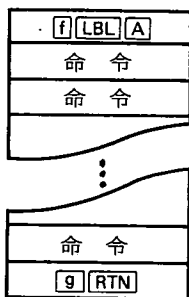
し、それから $\boxed{g} \boxed{x^2}$ を押します。次に π を表示させるために $\boxed{f} \boxed{\pi}$ を押します。最後に r^2 と π を掛けるために \boxed{x} を押して答を出します。このキー操作順序をまとめると下のようになります。

$\boxed{g} \boxed{x^2}$
 $\boxed{f} \boxed{\pi}$
 \boxed{x}

プログラムの始めと終り

プログラムの始め部分を明確にするために、 $\boxed{f} \boxed{LBL} \boxed{A}$ (label) キーと A~E の文字か 0~9 の数字キーを押してラベル名を付けます。計算機中に別々のプログラムを幾つか入れておいたり、一つのプログラムを複数の副プログラムに区分しておいても、ラベル符号やラベル番号が重複しなければどのようにでも使いわけができます。

プログラム・メモリー



\boxed{LBL} からプログラムが始まる。

\boxed{RTN} がプログラムの終り。

プログラムの終り部分を明確にするために $\boxed{g} \boxed{RTN}$ (return) 命令を使います。プログラム走行中に \boxed{RTN} 命令に出会うとすぐにライン 000 に戻ってプログラムを止めます。

注：プログラム計算中にそのときのプログラム・メモリーの終りまで行くと、 $\boxed{g} \boxed{RTN}$ に出会ったのと同じ効果があります。これは入れた(記憶させた)最後のプログラム命令の次に $\boxed{g} \boxed{RTN}$ が入っているのと同じなので、1ライン分命令が少なくて済みます。

完成したプログラム

円の半径を入力して円の面積を計算するプログラムがこれで完成しました。

- f** **LBL** **A** Aというプログラムの始めを表わす。
g **x²** 入力した半径の二乗（スタック内の数値の上昇や下降はない）。
f **π** π を呼び出す（スタック上昇）。
x 半径の二乗と π を掛け（スタック下降）て答を表示する。
g **RTN** プログラムの終り、ライン 000 に戻って停止する。

プログラムの入力

プログラムはプログラム・メモリー中に既に入れてあるプログラムの前か後に入れる必要があります。既に入れてあるプログラムの前に新しいプログラムを入れる（ライン 000 に行ってから新しいプログラム命令をキーインする）と、新プログラムの命令を一つ入れるごとに既に入れてあるプログラムが一つずつ後に下がります。

円の面積を計算するプログラムを入れる準備をしましょう。

1. **g** **P/R**を押して PRGM モードに切替える。表示に PRGM の文字が見える。
2. 前に入れておいたプログラムが不要なら **f** **CLEAR** **PRGM**を押して消す。（前に入れておいたプログラムが必要なら **f** **CLEAR** **PRGM** でなく **GTO** **000** と押す。 **GTO** **000** と押すとプログラム・メモリーの内容に無関係にライン 000 へ行ける。）

表示の左端に 000- が見えて、プログラム・メモリーの始めになったのが確認できるはずです。

円の面積のプログラムを次の順序でキーインして入れます。

f **LBL** **A**
g **x²**
f **π**
x
g **RTN**

まずプログラムの一番始めの **f** キーを押しましょう。

86 第5章 プログラム作成の基礎知識

キー操作 表 示

f **000-**

表示は何も変わっていません。一つのプログラム命令分のキー操作が終るまでは表示が変化しないからです。一番始めのプログラム命令の残りのキーを押してみましょう。

キー操作 表 示

LBL **000-**
A **001-42,21,11** プログラム・メモリーに
f **LBL** **A** が入りました。

プログラムのライン番号とキーコードを表示したら、そのプログラム命令のキー操作が終ったことを意味します。一つのプログラム命令をキーインするには命令によってキーを1～3回押す必要があることを覚えておいてください。

次のようにキーを押して残りのプログラム命令を入れてください。

キー操作 表 示

g **x²** **002- 43 11**
f **π** **003- 42 16**
x **004- 20**
g **RTN** **005- 43 32**

これで円の半径を入力して円の面積を計算するプログラムを HP-11C に入れ終りました。

プログラム計算

計算モードのときしかプログラム計算ができません。今入れた面積計算のプログラムを走らせるために **g** **P/R** を押して HP-11C を計算モードに切替えましょう。

計算するにはまず数値をキーインしてから、 **f** とプログラムのラベ

ル符号のキー（**A**～**E**）を押します。円の面積計算では半径を入れて **f****A** を押します。

例題 半径が7.5 cm, 9 インチ, 15.3 m の円の面積を計算してください。

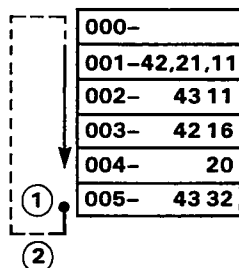
キー操作	表示	
7.5 f A	176.7146	cm ²
9 f A	254.4690	平方インチ
15.3 f A	735.4154	m ²

ここで計算機がどのように **LBL** を探すか説明しておきます。HP-11C を計算モードにしたときはまだプログラム・メモリーのライン 005(先程入れたプログラム命令の最後の部分) にセットしたままです。 **f****A** を押すとプログラム・メモリー中の 005 ラインから下の方(ライン番号が多い方)へ順番に **LBL** **A** の命令を探します。これだけではまだプログラム計算が始まりません。

それは次の二つの理由です。

1. 005 ライン以降に **LBL** **A** の命令がない
2. プログラム・メモリー中にもう命令がない。

そこで HP-11C は 000 ラインに戻ってまたプログラム・メモリー中をまた下の方に向かって探します。001 ラインで **f** **LBL** **A** を見つけそこからプログラム計算を始めます。



現在のプログラム・
メモリーの終り

プログラム命令の実行 計算機はプログラム命令をキーインした順番通りに、ここでは 002 ラインの **g** **x²**, 003 ラインの **f** **π** というように、**g** **RTN** か **R/S** (run/stop) 命令に出会うまで、またはそのときのプログラム命令の最後に到達するまで順々に実行します。この例では 005 ラインに **g** **RTN** があるので 000 ラインに戻って停止し

88 第5章 プログラム作成の基礎知識

ます。計算の答はXレジスタにあつてそれを表示しています。プログラム計算に時間がかかるときは計算している間 **running** の表示を点滅します。

プログラムできない機能 HP-11Cが PRGM モードのとき（表示に PRGM の文字が見えます）に大部分のキーはプログラム命令としてプログラム・メモリーに記録できますが、次の 10 種類だけはプログラム命令にはなりません。

f CLEAR	PRGM	g P/R	SST
f CLEAR	PREFIX	g MEM	←
GTO	nnn	./ ON	g BST
f USER			

USER モードでの操作

今度は HP-11C を USER モードに切替えて前回と同じ例題で円の面積を計算してみましょう。USER モードにしたのでキー操作がちょっと変わります。

キー操作

表示

f USER

USER モードに切替えたので
USER の文字が見える。

7.5 **A**

176.7146

USER モードにすると **A** ~
E の文字がそのキーで最優先になる。

9 **A**

254.4690

15.3 **A**

735.4154

4 **f** \sqrt{x}

2.0000

1 **f** e^x

2.7183

USER モードでは一段目の左
側 5 個の演算キーは、先に **f**
を押してからそのキーを押す。

1 **f** 10^x

10.0000

2 ENTER

2.0000

8 **f** y^x

256.0000

□ 5 **f** $1/x$

2.0000

f USER

2.0000

USER モードでないようにす
る。

プログラムの停止と一時表示

プログラム作成のときに、プログラム計算の途中で止めて計算用の数

値を入れたいと考えたと思います。あるいはプログラム計算の途中でしばらくの間数値を表示しそれからまた計算を続行させたいと考えたと思います。[R/S] (run/stop) と [f] [PSE] (pause) の二つのキーがこの目的に使えます。

プログラム計算途中の停止

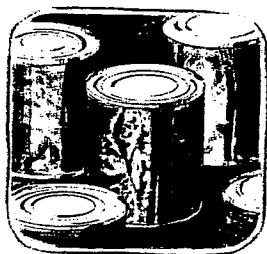
[R/S] (run/stop) はプログラム命令としても、単にキーを押しても使えます。

このキーを押すと

1. プログラム計算の途中であれば [R/S] を押すとプログラム計算を止める。
2. 計算モードでプログラム計算を止めたり・止まっているときに [R/S] を押すとプログラム計算を始めます。プログラム・メモリーの [R/S] 命令の次のラインから順に実行します。(計算モードで [R/S] を押し続けるとそのときのライン番号とキーコードを表示し、キーを放すとそのラインから実行を始めます。)

この [R/S] 命令をプログラムの途中で数値をキーインしたい部分に入れておけばよいわけです。数値をキーインしたら [R/S] キーを押してプログラムを再スタートさせます。

例題 ある製缶会社の技師が各種の円筒形の缶の体積を計算しようとしています。計算の途中で缶の底面積をノートに写したいと考えました。



次のプログラムはまず缶の底面積を表示して停止します。その答をノートに書き写したら計算を続行して缶の体積を計算します。計算式は次の通りです。

$$\text{体積} = \text{底面積} \times \text{高さ} = \pi r^2 \times h$$

90 第5章 プログラム作成の基礎知識

半径 r と高さ h は計算開始前に X と Y レジスタに入れておくものとします。

キー操作	表示	
f CLEAR PRGM	000-	プログラム・メモリーをクリアすると 000 ラインを表示する。
f LBL A	001-42,21,11	
g x²	002- 43 11	半径を二乗する。
f π	003- 42 16	π を呼び出す。
x	004- 20	底面積を計算する。
R/S	005- 31	底面積を記録するために停止
x	006- 20	体積の計算。
g RTN	007- 43 32	

HP-11C を計算モードに戻します。下表の空欄を計算してください。

高さ	半径	底面積	体積
25	10.0	?	?
8	4.5	?	?

キー操作	表示	
25 ENTER	25.0000	高さを Y レジスタに入れる。
10 f A	314.1593	半径を X レジスタに入れて計算を開始。底面積を計算してそれを表示したまま停止。
R/S	7,853.9816	最初の缶の体積を表示。
8 ENTER	8.0000	高さを Y レジスタに入れる。

4.5 **f** **A**

63.6173

半径を Xレジスタに入れて計算を開始。底面積を計算してそれを表示したまま停止。

R/S

508.9380

二番目の缶の体積を表示。

計算モードで高さを Yレジスタに、半径を Xレジスタに入れて **f** **A** を押すと底面積を計算したところで **R/S** 命令に出会って停止します。**R/S** を押すと体積を計算して 000 ラインに戻って停止します。

プログラム計算の途中で数値を入れる方法についてはこの本の第3部の数値入力 (212 ページ)、ユーザー指定キー (217 ページ) をご覧ください。

プログラム計算途中の一時表示

プログラム中に **f** **PSE** 命令があるとそこでプログラムが一時止まって答を表示し、それからまたプログラムを続行します。一時止まっている時間は約 1 秒間ですが、**f** **PSE** を繰返して使えばそれだけ一時停止の時間が長くなります。

f **PSE** の働きを見るために先程の円筒形の体積の計算プログラムを変更してみましょう。次のプログラムは体積計算の前に底面積を約 1 秒間だけ表示します。また例題を計算してどう違うかよく見てください。

プログラムを入れるために HP-11C を PRGM モードに切替えてください。**f** **CLEAR** **PRGM** を押してプログラム・メモリーをクリアして 000- を表示するようにします。次のようにプログラムをキーインしてください。

キー操作 表示

f **CLEAR** **PRGM**

000-

f **LBL** **A**

001-42,21,11

g **x²**

002- 43 11 半径を二乗する。

- f** **π** **003-** **42 16** π をXレジスタに呼出す。
x **004-** **20** 底面積を計算する。

f **PSE** **005-** **42 31** 底面積を1秒間だけ表示する。

x **006-** **20** 缶の体積を計算する。

g **RTN** **007-** **43 32**

このプログラムも高さをYレジスタに、半径をXレジスタに入力して計算するようにしました。プログラムを全部入れたらHP-11Cを計算モードにしてください。新しいプログラムで次表を完成させてみましょう。

高さ	半径	底面積	体積
20	15	?	?
10	5	?	?

キー操作

表 示

- 20** **ENTER** **20.0000** 高さをYレジスタに入れる。

15 **f** **A** **706.8583** 半径をXレジスタに入れて計算を始める。缶の底面積を計算して約1秒間だけ表示する。

14,137.1669 体積を表示してプログラム計算を終了。

10 **ENTER** **10.0000** 第二の缶の高さをYレジスタに入れる。

5 **f** **A** **78.5398** 半径をXレジスタに入れて計算を始める。缶の底面積を計算して約1秒間だけ表示する。

785.3982 体積を表示してプログラム計算を終了。

予期しない計算途中の停止

プログラムに各種のミスがあるとプログラム計算中に停止してしまいます。どういうときにプログラム計算途中で停止するかを次に列記しておきます。

9 **RTN** の実行 サブルーチン中でなければプログラム中に **9** **RTN** があるとそこで 000 ラインに戻って停止してしまいます。

プログラム・メモリーの最後に到達 プログラム・メモリーの最後の命令が **GTO** , **GSB** , **RTN** や **R/S** でなかったり、サブルーチンの中でないときにそのときのプログラム・メモリーの最後まで到達すると直ちに 000 ラインに戻って停止します。

どれかのキーにさわった プログラム計算中にキーにさわると計算を停止します。計算機はプログラム計算中には数値入力するとき以外は停止しないように設計してあります。プログラム計算途中で X レジスタに数値を入れるために数字キーを押すとその数が計算機中に入って、直ちに次のプログラム命令を実行しないで停止してしまいます。

プログラムが停止してしまったら **R/S** を押せば計算を続行しますが正しい答になるとは限りませんので、計算途中では数値入力のために停止したとき以外はキーにさわらないようにしてください。

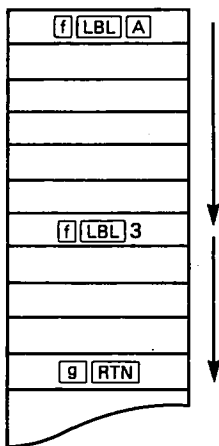
エラーによる停止 プログラム計算途中にエラーが発生したら（詳しくは付録 A エラー発生原因 219 ページを参照）直ちに計算を中止して停止し、Error と数字を表示します。エラーが発生したプログラムのライン番号とキーコードを見たいときは、まずどれかのキーを押してエラー表示を消しそれから **9** **P/R** を押して PRGM モードに切替えてください。

もしレジスタの直接演算をしてレジスタがオーバーフローした（計算機の範囲外まで大きくなった）ときは Error 1 を表示して停止します。しかしこの数値のためにそのレジスタ内容が変化することはありません。エラー表示を消せば、エラー発生前の数値を表示します。

もし計算した結果が $1.000000000 \times 10^{-99}$ よりも小さくなるとその数の代りに 0 を使って計算を続けます。これをアンダーフローと言います。

ラベル

前に説明しましたようにラベル (label) はプログラム中でどの部分からプログラム計算を始めるかという目印の役目をします。プログラム計算途中にラベルがあってもそれを無視して下の方に計算を続けます。例えば次のようなプログラムがあるときに **f** **[A]** を押すと、**f** **[LBL]** **[A]** からプログラム・メモリー中を下の方にプログラム計算を始めますが、**f** **[LBL]** **3** はそのまま素通りして **[RTN]** までくると 000 ラインに戻って停止します。



f **[A]** を押すとここから計算を始める。

これは **g** **[RTN]** や **[R/S]** ではないので **f** **[LBL]** **3** は無関係に(無視して)計算を続ける。

計算を続けて **[RTN]** まできたら 000 ラインに戻って停止する。

練習問題

- 次の式を使ってせっし (Celsius) の温度をかし (Fahrenheit) の温度に換算するプログラムを作って入力してください。プログラムの始めと終りは **f** **[LBL]** **[A]** と **g** **[RTN]** にして、 -40°C 、 0°C 、 72°C が $^{\circ}\text{F}$ になるか求めてください。

$$F = 1.8C + 32$$

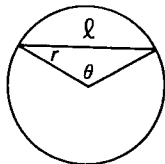
答 -40.0000°F , 32.0000°F , 161.6000°F .

2. 次の式で円の半径 r と角度 θ を入力して弦 l の長さを求めるプログラムを作ってください。

$$l = 2r \sin \frac{\theta}{2}$$

r , θ の順に入力するように工夫してみましょう。[f] [LBL] [B] から始めるようにして下表を完成してください。

r (m)	θ	l
25	30	?
50	45	?
100	90	?



答 12.9410m, 38.2683m, 141.4214m

3. 次のキーのキーコードは何でしょうか? [g] [%], [f] [→RAD], [STO] [+], 1, [SIN], [f] [LBL] [D], [STO] [RAN#].

答 43 14; 42 3; 44, 40, 1; 23; 42, 21, 14; 44 36*.

4. 次はどれもプログラムの一部ですが、それぞれプログラム・メモリーを何ライン使うでしょうか?

- a. 2 [ENTER] 3 [+].
 b. 10 [STO] 6 [RCL] 5 [x].
 c. 100 [STO] 1 50 [STO] [x] 1 [RCL] 2 [f] [π] [÷].

答 a, 4; b, 5; c, 10.

第6章 プログラムの編集

編集の要点

沢山プログラムを作った人でもプログラムのミスをする場合があります。このミスの原因は元の計算式を間違えたのから、計算機にプログラムを入れるときに間違えたのまで各種あると思います。ミスがあったらそれを見つけて直す必要があります。HP-11Cはミス発見が簡単にできるように設計してあります。

プログラムは数値メモリーの指定や計算機の状態（フラグや角度の単位など）が正しくないとやはりエラーを発生することがあります。この種のミスはプログラムを入れる前に不揮発性メモリーをリセットすればかなり防げます。またプログラム計算を始める前に手で状態指定するのも一つの方法です。もう一つの方法はプログラムの始めの部分にこの種の命令を入れておくことです。

プログラム・エラーの発見

プログラムをチェックする一番簡単で良い方法は答がいくつになるかわかっている数値を入れ、計算し答があっているかを確認することです。できれば大きな数値や小さな数値を入れて計算可能限度や計算誤差を調べておくといよいでしょう。場合によっては正しくない数値を入れて調べてみることも必要です。プログラム中に状態を指定する命令を入れたプログラムでは全部の数値レジスタに意味のない数値を入れてから角度の単位指定を変えたり、フラグをセットしたりクリアして何回も計算してみると安心です。

HP-11Cはプログラム・メモリーのどの部分も簡単に調べられ、編集（削除や追加）や間違いの点検が手軽にできるように設計してあります。プログラム計算中にエラーやオーバーフローで計算が止まったら、エラー表示を消せば停止前のXレジスタの値を表

示し、PRGM モードに切替えればプログラムのそのライン番号とキーコードが見られます。それだけでなくプログラムのどこで計算ミスが発生するのかわからなければ、プログラム命令を一つずつ実行して調べられるので、原因を探して訂正できます。

編集機能

HP-11Cの編集用に次の四種のプログラムできない機能があるので、プログラムの修正や変更操作が簡単にできます。

SST **BST** **GTO** **nnn** **↔**

SST (single-step : シングル・ステップ)

PRGM モードのとき

SST キーを押して放すとプログラム・メモリーの次のラインへ移動します。**SST** を押したままにすると **SST** を放すまでそのときのプログラム・メモリー中をどんどん下の方へ移動します。**SST** を押してもプログラム命令を実行することはありません。

000-

SST

001-42,21,11

計算モードのとき

SST を押すとプログラム・メモリーの次のラインへ移動し、キーを放すまでプログラムのライン番号とキーコードを表示します。**SST** を放すと直ちにそのラインの命令を実行します。

BST (back step : バックステップ)

g **BST** を押すと **SST** のときと同じようにプログラム・メモリー中を一つずつまたは続けて逆戻りします。(ただしプログラム命令を実行することはありません。)

002- 43 13

g **BST**

001-42,21,11

GTO **nnn** (go to line nnn : nnn ラインへ行け) PRGM モ

ードや計算モードで **GTO** \square nnn と押すと nnn と指定したプログラム・ラインへ行くことができます。(計算モードでは **GTO** \square nnn と任意のプログラム・ラインへ行くのと同じように、**GTO** の後に A~E または 0~9 のラベルを指定してそのラベルへ行くことができます。)

☒ (削除) PRGM モードで **☒** を押すとプログラム・メモリー中のそのときに表示しているプログラム命令を削除します。するとそれ以降の全部のプログラム命令が自動的に 1 ラインずつ繰上りになって、ライン番号も 1 ずつ少なくなります。計算モードで **☒** を押してもプログラム・メモリーは全然変化しないで、そのときに表示している Xレジスタの内容が 0 になるだけです (17 ページ表示のクリア: **CLx** と **☒** 参照)。

プログラム・メモリー例

☒ を押す前	→	☒ を押した後
001-42,21,11	→	001-42,21,11 (表示)
(表示) 002- 43 13	→	002- 23
003- 23	→	003- 45 0
004- 45 0	→	004- 45 1
005- 45 1	→	

プログラム編集の例

プログラム編集のときにどうなるかを見るために、ピタゴラスの定理を使って直角三角形の二辺 a と b を入れて斜辺 c を求めるプログラムを入れてみましょう。計算式は $c = \sqrt{a^2 + b^2}$ です。 a の値を Y; b の値を Xレジスタに入れてから計算を始めるようにしたプログラムです。



まず HP-11C を PRGM モードにしてください。

キー操作

f CLEAR PRGM

f LBL E

g x²

x ≧ y

g x²

+

√x

g RTN

表示

000-

001-42,21,15

002- 43 11

003- 34

004- 43 11

005- 40

006- 11

007- 43 32

プログラム・メモリをクリア。

プログラムのラベル。

 b を二乗。表示している Xレジスタの b^2 と Yレジスタの a とを入替えて、 a を表示させる。 a を二乗。 $(a^2 + b^2)$ 。 $\sqrt{(a^2 + b^2)}$ 。

プログラムの終り。

HP-11C を計算モードに戻しましょう。

プログラムを試めすために a を 22 m, b を 9 m として斜辺を求めてみます。

キー操作

22 ENTER

9

f E

表示

22.0000

9

23.7697

 a をキーイン。 b をキーイン。

求める斜辺の長さ (m)。

プログラムの 1 ステップずつの計算

長いプログラムのどこか一つでも間違えていると正しい答が求まりません。このようなときには計算モードで **SST** を使って順々に計算しながらチェックできます。

1 ステップずつの計算は計算機のそのときの位置から始まります。プログラムを走らせた後ならプログラムの最後の **RTN** で 000 ラインに戻っているはずですから、始めに入れる数値をキーインすれば直ちに 1 ステップずつ計算を始められます。 **SST** を押したままにするとこれから実行するプログラム命令のライン番号とキーコードを表示します。 **SST** を放すとその命令を実行します。

キー操作	表示	
22 ENTER	22.0000	Yレジスタに a を入れる。
9	9	Xレジスタに b を入れる。
SST	001-42,21,15	SST を押している間は 001 ライン (f LBL E がある) のキーコードを表示。
	9.0000	SST を放すと 001 ラインを 実行 (ラベルなので変化しない)。
SST	002- 43 1	g x² のキーコード。
	81.0000	SST を放すと x² を実行。
SST	003- 34	x\geqy のキーコード。
	22.0000	SST を放すと x\geqy を実行。
SST	004- 43 11	
	484.0000	x² を実行。
SST	005- 40	
	565.0000	+ を実行。
SST	006- 11	
	23.7697	\sqrt{x} を実行。
SST	007- 43 32	
	23.7697	g RTN を実行。プログラム はここで終り、000 ラインに 戻って停止。

注：プログラム・メモリー中にプログラム命令がなければ **SST** を押しても先に進みません。プログラム・メモリー中の最後のラインが **GTO** (go to : ジャンプ) か **GSB** (go to subroutine : サブルーチンへジャンプ) でなければ、**SST** を押すと 000 ラインへ戻ります。最後の命令が **GTO** か **GSB** だとその **GTO** か **GSB** を実行してプログラム・メモリー中の指定の場所へジャンプします。

PRGMモードでの **SST** と **BST**

プログラム・メモリー中を1ラインずつ動きます。PRGMモードで

は **SST** を押して放すと、プログラム・メモリー中の次のラインへ動いてその命令を表示します。しかし命令は実行しません。(**g** **BST**) は同じように働きますが1ラインだけ逆戻りしてその命令を表示します。)

キー操作

表示

g **P/R**

000-

HP-11C を PRGM モードにする。

SST

001-42,21,15

SST

002- 43 11

SST

003- 34

SST

004- 43 11

SST

005- 40

SST

006- 11

SST

007- 43 32

g **P/R**

HP-11C を計算モードにする。

プログラム・メモリー中の連続移動 PRGM モードで **SST** や **BST** を押したままにするとプログラム・メモリー中を前や後へ連続移動します。(押したときはそのラインを約2秒表示し、次からは約0.5秒ずつ表示します。)

プログラムの変更

HP-11C の編集機能を見てもらうために、99ページのピタゴラスの定理のプログラムを変更して任意の時点でXレジスタの内容を自動的に表示するようにしてみましょう。それには次のプログラムの矢印のところに **f** **PSE** (pause: 一時停止) を追加すればよいわけです。

f **LBL** **E**

001-42,21,15

g **x²**

002- 43 11

x **≧** **y**

003- 34

g **x²**

004- 43 11

+

005- 40

√ **x**

006- 11

g **RTN**

007- 43 32

この三つの命令の後に
f **PSE** を一つずつ追
加する。

命令の追加 新しいプログラム命令は000ラインからそのときのプログラム・メモリー中の命令があるところまでならどこにでも追加できます。次のようにして追加します。

102 第6章 プログラムの編集

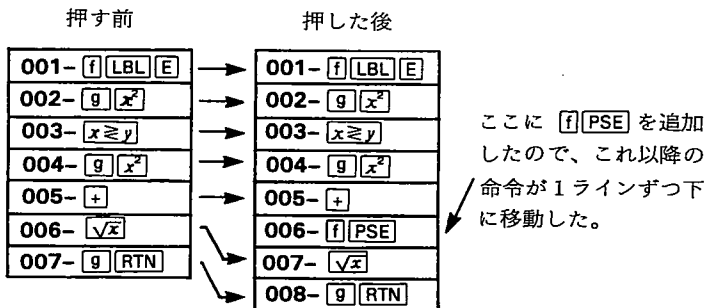
1. 計算機を PRGM モードにする。
2. プログラム・メモリー中の追加したい前のラインを表示させる。
3. 新しい命令をキーインする。こうするとプログラム・メモリー
の次のラインに新しい命令が入ります。そしてそれ以降の命
令は自動的に1ラインずつ下に移動してライン番号も変ります。

新しい命令を一つ追加するたびにそれ以降のプログラム命令のライン
番号がずれるので、追加するには 000 ラインから一番遠いところから
始めて、順々に 000 ラインの方へ戻るようにするとよいでしょう。

例 **[+]** 命令の後に **[PSE]** 命令を追加しましょう。

キー操作	表示	
[GTO] [0] 000		000 ラインへ行く。
[g] [P/R]	000-	000 ライン。
[SST]	001-42,21,15	005 ラインへ行く。
[SST]	002- 43 11	
[SST]	003- 34	
[SST]	004- 43 11	
[SST]	005- 40	005 ライン。
[f] [PSE]	006- 42 31	006 ラインに [PSE] 命令を 追加した。

005 ラインに行って **[f]** **[PSE]** を押すとプログラム・メモリーは次のよ
うに変化します。



任意のライン番号へ行く。このように **SST** や **g** **BST** を使ってプログラム・メモリー中を1ラインずつや、続けて移動する方法は操作は簡単ですが時間がかかります。先程 000 ラインへ行くのに **GTO** **.** nnn とした方法を使えば直ちに nnn ラインへ行けます。**GTO** **.** と押してプログラム・メモリー中のライン番号を3桁で指定するだけで直ちにそのライン番号へ行きます。(この **GTO** **.** nnn は PRGM モードでも計算モードでも操作できますが、途中のプログラム命令は実行しません。) ここでは2番目の **g** **x²** の後に **PSE** を入れるために 004 ラインへ行きましょう。

キー操作

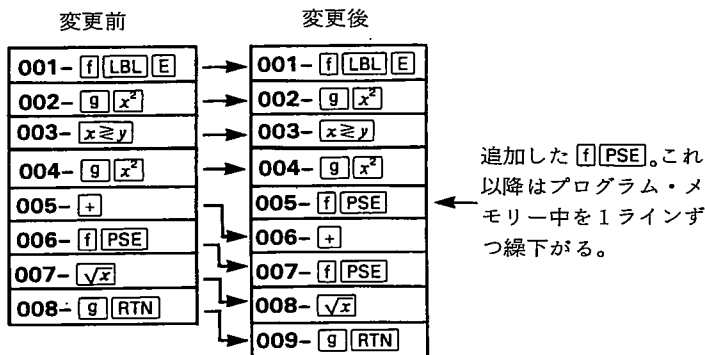
GTO **.** 004**f** **PSE**

表示

004- 43 11 004 ラインへ。

005- 42 31 004 ラインの次に **PSE** を追加。

004 ラインの後に **f** **PSE** を追加すると、プログラム・メモリーは次のように変化します。



今度は **BST** を使って3番目の **fPSE** を追加しましょう。

キー操作

表示

g **BST**

005- 42 31 プログラム・メモリーのその
ときの位置。

g **BST**

004- 43 11

g **BST**

003- 34

f **PSE**

002- 43 11 このラインの次に **PSE** を追
加する。

003- 42 31 **PSE** を追加したので 004 ラ
イン以降の全部の命令が1ラ
インずつ繰下がる。

変更後のプログラムは次のようになっているはずです。できれば **SST** を押してプログラムがこのようになっているか調べてください。

キー操作

表示

f **LBL** **E**

001-42,21,15

g x^2

002- 43 11

f **PSE**

003- 42 31

 $x \geq y$

004- 34

g x^2

005- 43 11

f **PSE**

006- 42 31

+

007- 40

f **PSE**

008- 42 31

 \sqrt{x}

009- 11

g **RTN**

010- 43 32

もう一度プログラムを走らせてみましょう。前回と同じく a が 22 で b が 9 として計算します。

キー操作	表示	
g P/R		計算モードにする
22 ENTER	22.0000	a の値を Y レジスタに入れる。
9	9	b の値を X レジスタに入れる。
f E	81.0000	003 ラインの $\boxed{x^2}$ (b^2) の答を一時表示する。
	484.0000	006 ラインの $\boxed{x^2}$ (a^2) の答を一時表示する。
	565.0000	007 ラインの $\boxed{+}$ (a^2+b^2) の答を一時表示する。
	23.7697	最後の答の c ($c = \sqrt{a^2+b^2}$)。

長いプログラムのときの命令の追加 プログラム全体で 203 ラインになっているときにはプログラム命令の追加ができません。このときに命令をキーインすると Error 4 を表示してプログラムの変更を受け付けません。(付録 C メモリーの自動配分 214 ページを見てください。)

命令の削除 プログラム・メモリー中の命令を削除したいことがあると思います。それには HP-11C を PRGM モードにして **GTO**、**SST** や **BST** を使って削除したい命令を表示させ、 \leftarrow を押して削除します。(\leftarrow を使ってプログラム・メモリー中の命令を削除すると、それ以降の命令は 1 ラインずつ繰上りがあってライン番号も自動的に変わります。削除するとその命令の一つ前にあった命令を表示します。)

例 今までやったピタゴラスの定理のプログラムで **f** **PSE** を一つだけ(二乗の和を表示する部分)を残して、二つの **PSE** (003 と 006 ライン)を削除してみましょう。

キー操作

表示

[G][P/R]

000-

PRGM モードにする。

[GTO][]006

006-

42 31 006 ラインへ行く。

[←]

005-

43 11 [f][PSE] を削除したのでその前のライン (005 ライン) を表示する。この次以降の命令は一つずつ繰上がって自動的に番号が変わる。

[GTO][]003

003-

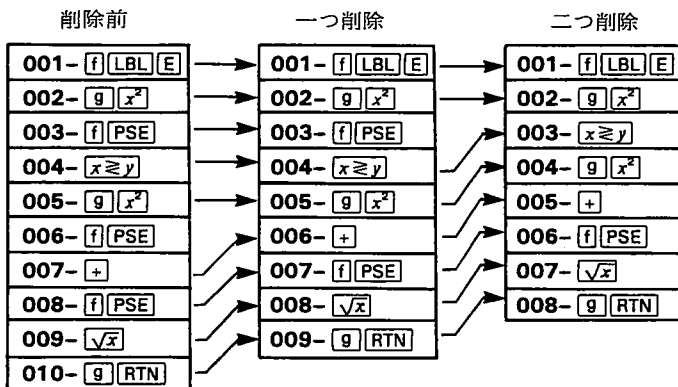
42 31 003 ラインへ行く。

[←]

002-

43 11 [f][PSE] を削除したのでその前のライン (002 ライン) を表示する。
この次以降の命令は一つずつ繰上がって自動的に番号が変わる。

003 と 006 ラインの [PSE] を削除するとどうなるかを図解すると次のようになります。



ピタゴラスの定理のプログラムをこのように変更すると、一時表示するのは二乗の和だけになります。次に斜辺の長さを計算して終ります。

今度は a が17m, b が34mのときの斜辺を計算してみましょう。

キー操作 表 示

g **P/R**

HP-11C を計算モードにする。

17 **ENTER** 34

34

f **E**

1,445.0000

辺 a と b の二乗の和を一時表示。

38.0132

斜辺の長さ。

64ライン以上の長いプログラムのときに命令を削除すると、場合によってはメモリーの自動配分が逆になります。例えばある71ラインのプログラムで命令を一つ削除すると、プログラムの71~78ライン用に使っていたメモリーが数値レジスタR₈として使えるようになります。(付録Cメモリーの自動配分をみてください。)

練習問題

1. 次のプログラムはある金融機関で定期預金(複利)の元利合計

$FV = PV(1+i)^n$ の計算に使っているものです。FVは元利合計、

PVは預金額, iは1期間あたり

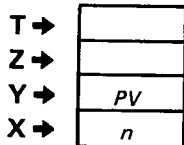
の利率で小数で入力, nは期間

数です。利子計算は年1回で、

利率が年7.5%, 右図のように

PVとnをスタックに入れてお

いてから計算するものとします。



キー操作

表 示

f **LBL** **A**

001-42,21,11

f **FIX** **2**

002-42, 7, 2

1	003-	1	
\square	004-	48	
0	005-	0	利率
7	006-	7	
5	007-	5	
$x \geq y$	008-	34	
y^x	009-	14	$(1+i)^n$
x	010-	20	$PV(1+i)^n$
g RTN	011-	43 32	

- a. このプログラムを HP-11C に入れてください。
- b. 1,000ドルを5年間預けると幾らになるでしょうか。
(答1,435.63ドル)
2,300ドルを4年間預けると幾らになるでしょうか。
(答3,071.58ドル)
- c. このプログラムの利率を年7.5%から8%に変えてください。
- d. プログラムを変更したら500ドルを4年間, 2,000ドルを10年間預けるとそれぞれ幾らになるでしょうか。
(答680.24ドル; 4,317.85ドル)

2. 次のプログラムは高さをキーインして, そこから物体を落してから地面に届くまでの時間を計算するものです。(ただし空気の抵抗は無視してあります。) 高さ h (m) を Xレジスタにキーインして **[B]** を押すと次式の t (秒) を計算します。

$$t = \sqrt{\frac{2h}{9.8}}$$

- a. HP-11C に入っているプログラムを全部消し, **[FIX]** 4表示に戻して次のプログラムを入れてください。

キー操作

f LBL B

2

x

9

□

8

÷

√x

g RTN

表 示

001-42,21,12

002- 2

003- 20

004- 9

005- 48

006- 8

007- 10

008- 11

009- 43 32

- b. パリのエッヘル塔（高さ300.51m）と、地上1,000mの気球から小石を落とすと地面まで何秒かかるでしょうか。（答7.8313秒；14.2857秒ただし小石でも人間にぶつくと殺人事件になります。）
- c. 今度は次式のように高さをフィートでキーインするようにしてください。

$$t = \sqrt{\frac{2h}{32.1740}}$$

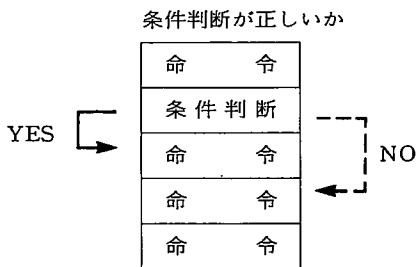
- d. 変更後のプログラムで高さ550フィートのグランド・クーリー・ダム、ニューヨークの世界貿易センタービル（高さ1350フィート）から小石を落したときの時間を計算してください。（答5.8471秒；9.1607秒）

第7章 プログラムの条件判断と コントロール

条件判断とコントロールの要点

プログラム中の条件判断

HP-11Cにはプログラム中の条件判断（大小の比較）用に8種あります。プログラム計算途中に条件でYESになれば条件判断の次の命令を実行します。条件判断でNOになれば条件判断の次の命令だけを飛び越してその次の命令を実行します。



HP-11Cの条件判断は次のようにXレジスタの値とYレジスタの値を比較するものと、Xレジスタの値と0とを比較するものがあります：

- $x \leq y$ Xレジスタの値がYレジスタの値以下かどうか調べて判断する。
- $x > y$ Xレジスタの値がYレジスタの値よりも大きいかどうか調べて判断する。
- $x \neq y$ Xレジスタの値とYレジスタの値が等しくないかどうか調べて判断する。

- f** **x=y** Xレジスタの値とYレジスタの値が等しいかどうか調べて判断する。
- g** **x<0** Xレジスタの値が0よりも小さいかどうか調べて判断する。
- g** **x>0** Xレジスタの値が0よりも大きいかどうか調べて判断する。
- g** **x≠0** Xレジスタの値が0と等しくないかどうか調べて判断する。
- g** **x=0** Xレジスタの値が0と等しいかどうか調べて判断する。

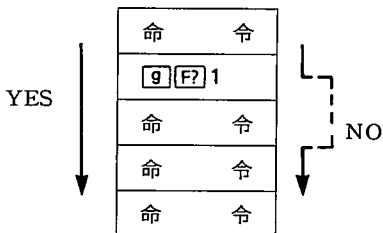
フラグ

プログラム中のもう一つの条件判断用としてフラグがあります。フラグはセットしてある、クリアしてあるのどちらかの状態だけです。プログラム中でこのフラグがセットしてあるかどうかの判断をするわけです。プログラム中のフラグ判断も大小判断のときと同じような動作をします。

HP-11Cのフラグは0と1の二つあります。フラグをセットするには **g** **SF** (set flag) と押してから0か1の数字キーを押します。フラグをクリアするには **g** **CF** (clear flag) と押してから0か1の数字キーを押します。フラグがセットしてあるかどうか判断するには **g** **F?** (is flag set?) と押してから0か1の数字キーを押します。

g **SF** *n* でフラグをセットすると **g** **CF** *n* でクリアするか不揮発性メモリーをリセットするまでセットしたままになります。

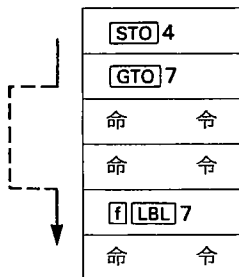
フラグ1がセットしてあるか



プログラムの流れのコントロール

GoTo (……へ行け, ……へジャンプ)

GTO はプログラムの内容点検や編集に使いましたが、プログラム・メモリー中に入れてプログラムの流れ(実行順序)を変えるのにも使えます。プログラム中では **GTO** の後に **A** ~ **E** か 0 ~ 9 のラベルを指定します。



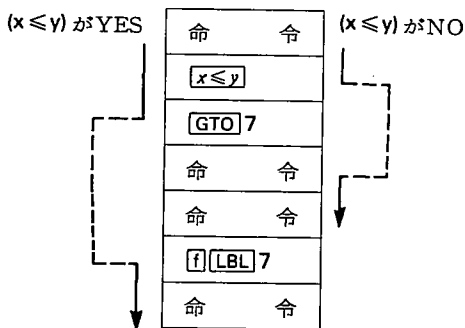
プログラム計算中に **GTO** ラベルがあると一たん計算を保留して、メモリー中を下の方に指定したラベルを探しはじめます。そのラベルが見つかるとそこにジャンプしてまた計算を再開します。

ジャンプとループ

無条件ジャンプ 無条件ジャンプは単に **GTO** を入れておいて、プログラムがそこまで進行したら数値などに無関係に別の所にジャンプさせることができます。この **GTO** を無条件ジャンプといいます。

条件ジャンプ 次の図のように条件判断(フラグを含む)と **GTO** を組合せると、その **GTO** を条件ジャンプといい、**GTO** で指定したラベルへジャンプします。つまり条件判断の答によって **GTO** を実行するかしないかが決まります。

条件ジャンプ

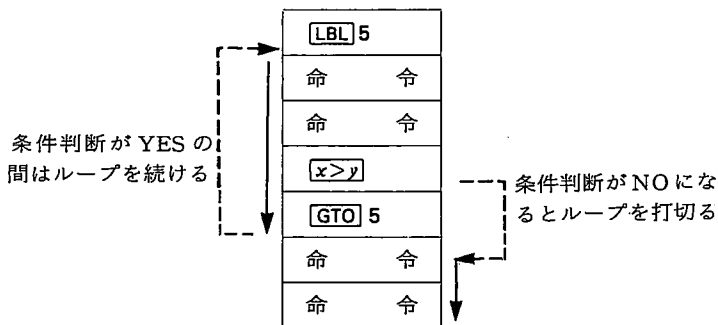


HP-11Cが計算モードのときに **GTO** とラベルのキーを押すと直ちにその指定したラベルへジャンプして、そこで止まっています。この機能は特定のラベルのプログラム命令をチェックしたり変更するとき特に便利です。

ループ ループはジャンプの特殊な場合で、**GTO** を使って一連の命令を何回か繰返すことです。ループのときにはカウンタを使って計算の繰返し回数を制御（コントロール）させる方法をよく使います。このときにはループの途中でカウンタの条件判断を使ってループを続けるか終了するか判断をさせるのが普通です。

次の図は **GTO** の前に条件判断を使ってループを続けるかどうかの判断をさせている例です。ループの途中で条件判断の答が NO になれば **GTO** を飛びこしてそれ以降の命令を実行します。ループの途中で無条件ジャンプを使っているのを無限ループと言って、止める操作をするまで何回でもループを繰返します。

条件ジャンプ
(ループの制御用)



条件判断を使ったループ制御の例 次は1から10までの平方根を求めるプログラムです。整数の平方根を求めるたびに一時表示をして、 $x > y$ で10より小さいかどうかの条件判断をします。整数が10よりも小さければ条件判断の次の $GTO 0$ で1多い整数の平方根を計算します。整数が10になって条件判断の答がNOになると $GTO 0$ を飛びこすのでループ計算が終わりになります。(このプログラムのループ部分は004～015ラインです。)

キー操作

 \boxed{g} $\boxed{P/R}$

表示

000-

HP-11CをPRGMモードにする。

 \boxed{f} \boxed{CLEAR} \boxed{PRGM}

000-

 \boxed{f} \boxed{LBL} \boxed{C}

001-42,21,13

0

002-

0 } 前の計算に関係ないように

 \boxed{STO} 1

003-

44

1

}

R1をクリアする。

 \boxed{f} \boxed{LBL} 0

004-42,21, 0 ループの始め。

1

005-

1

 \boxed{STO} $\boxed{+}$ 1

006-

44,40,

1

整数のカウンターの増加。

RCL 1	007-	45 1	
f PSE	008-	42 31	増加した整数の一時表示。
\sqrt{x}	009-	11	整数の平方根の計算。
f PSE	010-	42 31	平方根の一時表示。
g LSTx	011-	43 36	ラスト x レジスタの整数を呼び出す。
1	012-	1	
0	013-	0	
f x>y	014-	42 20	ラスト x が 10 よりも小さいかどうか比較する。
GTO 0	015-	22 0	条件ジャンプ。前回の整数が 10 よりも小さかったら LBL 0 へ行く。
g CLx	016-	43 35	そうでなかったらループへ行
g RTN	017-	43 32	かずに表示用の X レジスタをクリアしてプログラムを止める。
g P/R			HP-11C を計算モードに戻す。

このプログラムを動かすには **f** **C** を押します。HP-11C は 1 から 10 までの整数とその平方根を順次表示します。0.0000 を表示したらプログラムの終わりです。

動作の説明 **f** **C** を押すと HP-11C はプログラム・メモリー中の **f** **LBL** **C** を探し始めます。このラベルが見つかりると 014 ラインの条件判断の **x>y** までの命令を順次実行します。**x>y** の答が YES 一つつまり整数が 10 よりも小さい一ときは次の **GTO** 0 のために 004 ラインの **LBL** 0 へジャンプして次のループ計算を続けます。**x>y** の答が NO 一つつまり計算に使った整数が 10 になったら **GTO** 0 を飛びこすのでループが終了します。

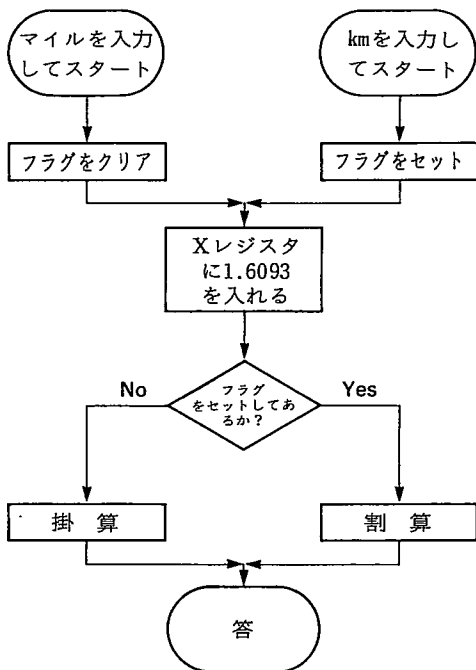
ループとループ制御の方法についてはこの本の第 3 部のループ (214 ページ) にも説明があります。

問題 販売手数料を計算して表示するプログラムを作ってください。売上げが 100 未満のときは手数料が 10%, 100 以上のときは 15% になります。条件判断を使って仕上げてください。

フラグの使用

x と y の比較や、 x と0の比較のようにフラグを使っても条件判断をさせることもできます。 x と y や、 x と0のように数値の大小などで比較しましたが、フラグは計算機の状態を調べます。つまりプログラムの途中にフラグのセット (**g** **SF** n) を入れておき、フラグはどうか (**g** **F?** n) によってジャンプしたりジャンプしなかったりに分けます。

例 次のプログラムはマイルからkm (**f** **A**), kmからマイル (**f** **B**) の換算用です。ここではフラグ0の状態によってkmを入力した(換算率で割る)か、マイルを入力した(換算率を掛ける)かを区分しています。



キー操作	表示	
g P/R	000-	HP-11C を PRGM モードにする。
f CLEAR PRGM	000-	プログラム・メモリをクリアする。
f LBL A	001-42,21,11	マイルを入力したときのプログラムの始め。
g CF 0	002-43, 5, 0	フラグ0をクリアする。(後ほどのためにマイルを入力したことを覚えさせておく。)
GTO 1	003- 22 1	LBL 1 へ無条件ジャンプ。
f LBL B	004-42,21,12	kmを入力したときのプログラムの始め。
g SF 0	005-43, 4, 0	フラグ0をセットする。(後ほどのためにkmを入力したことを覚えさせておく。)
f LBL 1	006-42,21, 1	ここから換算を始める。
1	007- 1	} 換算率を Xレジスタに入れる。
.	008- 48	
6	009- 6	
0	010- 0	
9	011- 9	
3	012- 3	
g F? 0	013-43, 6, 0	フラグの判断。フラグをセットしてあれば (kmの入力だった) LBL 2 へジャンプ。
GTO 2	014- 22 2	フラグをセットしなかったら (マイルの入力だった) kmを計算して停止。
x	015- 20	マイルを計算して停止。
g RTN	016- 43 32	HP-11C を計算モードにする。
f LBL 2	017-42,21, 2	
+	018- 10	
g RTN	019- 43 32	
g P/R		

プログラムを走らせて 26 マイルを km に、1.5 km をマイルに換算してみましょう。

キー操作

表示

26

26

マイルの値をキーイン。

f A

41.8418

マイルからkmへの換算。その
答を表示。

1.5

1.5

kmの値をキーイン。

f B

0.9321

kmからマイルへの換算。その
答を表示。

動作の説明 数値を入力し計算の目的によって **f A** (マイルからkmへ) か **f B** (kmからマイルへ) を押すと、フラグ0をクリアまたはセットしてそれからプログラム本体 (ラベル1以降) を実行します。換算率をXレジスタに入れてからフラグ0の判断をします。フラグをセット (YES) してあると **LBL** 2へジャンプし、マイルへの計算をして止まります。フラグをクリア (NO) してあると **GTO** 2を飛びこしてkmへの計算をして止まります。このフラグの判断によって二つのラベルのどちらからスタートしたかがわかるので目的の計算ができるわけです。

もっと詳しい説明はこの本の第3部のフラグ (215ページ) を見てください。

第 8 章 サブルーチン

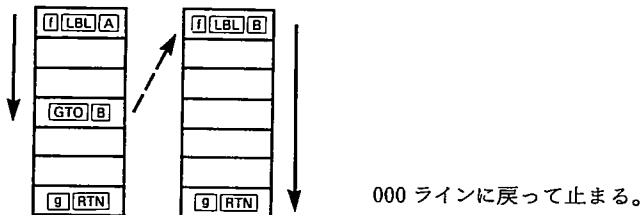
サブルーチンの要点

少し複雑なプログラムになると、同じ計算手順を別々のところで別々に使うことがあります。この一連の計算手順を一つのサブルーチンとしておいて、プログラム中の必要なところでこのサブルーチンを使うようにするとプログラムが短くなります。

サブルーチンへのジャンプ

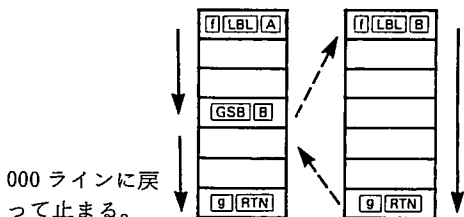
サブルーチンへジャンプさせるには **[GSB]** の後に目的のラベルの A～E または 0～9 を指定します。**[GSB]** は **[GTO]** のときと同じように指定したラベルへジャンプします。しかしプログラム中で **[GSB]** を実行するとそこで RTN (return : 戻る) 情報を一時保留します。RTN 情報を一時保留しているときに(サブルーチン先で) **[RTN]** に会うと直ちに元の **[GSB]** の次の命令に戻ります。そしてそこからプログラムを順次実行します。(**[GSB]** の次に戻ったときに一時保留の RTN 情報を自動的に解除します。(次の図は普通のジャンプと、サブルーチンのジャンプを比較したものです。

普通のジャンプ



- * **[GSB]** の後に **[A]** ～ **[E]** を押すときには **[f]** キーを押さなくてよい。78 ページのキーの省略をみてください。

サブルーチン



000 ラインに戻って止まる。

主プログラムの **GSB** の次に戻り、一時保留の **RTN** 情報を解除する。

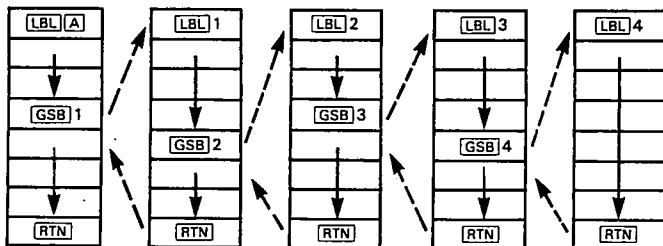
このように **GSB** (go to subroutine: サブルーチンへジャンプ) と **GTO** (go to branch: ジャンプ) の違いは **RTN** でどこに戻るかだけです。

GTO のときに **RTN** に出会うと 000 ラインに戻って止まり、**GSB** のときに **RTN** に出会うと主プログラムに戻って主プログラムを続行します。

サブルーチンの限度

サブルーチンの中から別のサブルーチンへジャンプさせることも可能です。これを「サブルーチンのネスティング (nesting)」とも、サブルーチンを重ねるともいって、これは HP-11C が同時に **RTN** 情報を幾つまで一時保留できるかによって決まります。サブルーチンを重ねたのを図解すると図のようになります。

主プログラム



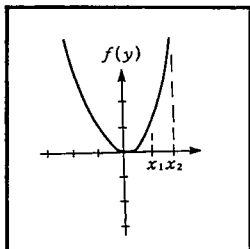
終り

図のようにサブルーチンが4段以内なら主プログラムに戻れます。サブルーチンを5段にすると、5段目のサブルーチン・ジャンプに出会うと計算を中止して Error 5 を表示します。

注：どんな場合でも一時にサブルーチンを重ねるのは4段以内ですが、重ねたサブルーチンの個所数や重ねていないサブルーチンの個数の制限はありません。

サブルーチンの使用法

例 図の $y=x^2-\sin x$ のグラフで、 x_1 と x_2 の間の勾配を計算するプログラムを作ってください。

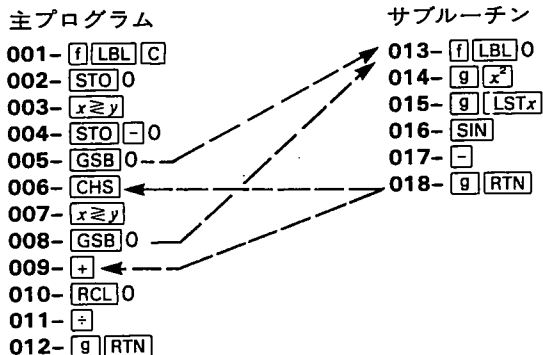


方法 この曲線の勾配の計算式は次の通りです。

$$\frac{y_2 - y_1}{x_2 - x_1} = \frac{(x_2^2 - \sin x_2) - (x_1^2 - \sin x_1)}{x_2 - x_1}$$

ここでは $x^2 - \sin x$ の計算を2回繰り返す必要があります（1回目は $x = x_1$ 、2回目は $x = x_2$ ）。

このように二つの x について同じ計算を2回しているのを、これを一つのサブルーチンとしてまとめてしまえばプログラムが短くてすみます。



([DEG] 単位になっているものとしてプログラムを作りました。)

Yレジスタに x_1 , Xレジスタに x_2 を入れてから [f] [C] のキーを押すと、001ラインの [LBL] [C] からプログラム命令を実行し始めます。005ラインの [GSB] 0 に出会うと直ちに013ラインの [LBL] 0 にジャンプして y_1 の計算を始めます。一例として x_1 が2, x_2 が3のときどんな計算をしているか見てみましょう。

	001	002	003	004
T →				
Z →				
Y →	2	2	3	3
X →	3	3	2	2

キー → [f] [LBL] [C] [STO] 0 $x \geq y$ [STO] [-] 0
 (Yに x_1 , X (x_2 を R0 (交換) (R0を $x_2 - x_1$ に x_2 がある) 入れる) にする)

	005	013	014	015
T →				
Z →				3
Y →	3	3	3	4
X →	2	2	4	2

キー → $\boxed{\text{GSB}}0$ $\boxed{f}\boxed{\text{LBL}}0$ $\boxed{g}\boxed{x^2}$ $\boxed{g}\boxed{\text{LST}x}$
 (ラベル0 (サブルーチン (x₁²) (x₁)
 にジャンプ) の始め)

	016	017	018	006
T →				
Z →	3			
Y →	4	3	3	3
X →	0.0349	3.9651	3.9651	-3.9651

キー → $\boxed{\text{SIN}}$ $\boxed{-}$ $\boxed{g}\boxed{\text{RTN}}$ $\boxed{\text{CHS}}$
 (sin x₁) (x₁² - sin x₁ (主プログラム (-x₁²
 = y₁) ムに戻る) - sin x₁))

018ラインでサブルーチンからもとの主プログラムに戻って先程の $\boxed{\text{GSB}}$ の次の命令から続行します。008ラインでまた $\boxed{\text{GSB}}0$ に出会うとまた013ラインの $\boxed{\text{LBL}}0$ にジャンプします。

	007	008	013	014
T →				
Z →				
Y →	-3.9651	-3.9651	-3.9651	-3.9651
X →	3	3	3	9

キー → $\boxed{x\geq y}$ $\boxed{\text{GSB}}0$ $\boxed{f}\boxed{\text{LBL}}0$ $\boxed{g}\boxed{x^2}$
 (交換) (ラベル0に(サブルーチン (x₂²)
 ジャンプ) の始め)

	015	016	017	018
T →				
Z →	-3.9651	-3.9651		
Y →	9	9	-3.9651	-3.9651
X →	3	0.0523	8.9477	8.9477

キー → $\boxed{g} \boxed{LSTx}$ \boxed{SIN} $\boxed{-}$ $\boxed{g} \boxed{RTN}$
 (x_2) $(\sin x_2)$ $(x_2^2 - \sin x_2^2)$ (主プログラ
 $= y_2)$ ムに戻る)

$\boxed{LBL} 0$ のサブルーチンで 2 回目に計算すると y_2 が求まり、018 ラインの $\boxed{f} \boxed{RTN}$ でもとのプログラムの $\boxed{GSB} 0$ の次の命令がある 009 ライン目に戻ります。このとき Xレジスタには y_2 、Yレジスタに y_1 があります。これ以降で勾配を計算します。

	009	010	011	012
T →				
Z →				
Y →		4.9826		
X →	4.9826	1	4.9826	4.9826

キー $\boxed{+}$ $\boxed{RCL} 0$ $\boxed{\div}$ $\boxed{g} \boxed{RTN}$
 $(y_2 - y_1)$ $(x_2 - x_1)$ $\left(\frac{y_2 - y_1}{x_2 - x_1} \right)$ (プログラ
ム終り)

計算が終って停止したときには x_1 と x_2 の間の勾配を表示しています。

それでは 122 ページの通りにプログラムをキーインして、 x_1 と x_2 が次の組合せのときの勾配を求めてみてください：0.5, 1.25 ; 2.52, 3.72 ; 5, 7。

答：1.7325, 6.2226, 11,9826

サブルーチンについてはこの本の第3部サブルーチン (210 ページ) でも説明いたします。

[GTO] や **[GSB]** のときはプログラム・メモリー中を下に向かって指定のラベルを探すということを覚えておいてください。これを知っていれば同じプログラム中に同じラベルを何回か重複して使うことも可能になります。

例 次のプログラムでは $\sqrt{x^2+y^2+z^2+t^2}$ の計算に主プログラムの始めとサブルーチンの始めに同じ**[LBL]** **[A]**を使っています。変数(キーインする数値) x, y, z, t をスタックに入れておいて **[A]** を押すという操作で計算するようにプログラムをまとめました。*

キー操作	表示	
[G] [P/R]	000-	
[f] [CLEAR] [PRGM]	000-	
[f] [LBL] [A]	001-42,21,11	
[G] [x²]	002- 43 11	x^2
[GSB] [A]	003- 32 11	y^2 を計算して x^2+y^2 を求める。
[GSB] [A]	004- 32 11	z^2 を計算して $x^2+y^2+z^2$ を求める。
[GSB] [A]	005- 32 11	t^2 を計算して $x^2+y^2+z^2+t^2$ を求める。
[√x]	006- 11	$\sqrt{x^2+y^2+z^2+t^2}$ の計算
[G] [RTN]	007- 43 32	
[f] [LBL] [A]	008-42,21,11	
$x \geq y$	009- 34	
[G] [x²]	010- 43 11	
[+]	011- 40	
[G] [RTN]	012- 43 32	
[G] [P/R]		

* しかし同じプログラム中で同じラベルを使うとわかりにくかったり、プログラムのミスをすることがあるので、次ページ以後はどうしても必要なとき以外は同じラベルを使わないようにしてあります。

次の変数のときの値を求めてみましょう。

$$x=4.3, y=7.9, z=1.3, t=8.0$$

キー操作	表示
8 ENTER	8.0000
1.3 ENTER	1.3000
7.9 ENTER 4.3 f A	12.1074



第9章 Iレジスタ

Iレジスタの要点

I (index : 指標, 目じるし) レジスタはHP-11Cのプログラム用として強力な道具です。単に数値をストアしておいて後で使うだけでなく, 次のようなことにも使えます。

- プログラムのループ・カウンタとコントロール機能
- 数値レジスタの間接指定, 間接ジャンプ, 間接サブルーチン・ジャンプ

ループ・コントロールや間接指定のときには前もってIレジスタにコントロール用の数値をストアしておく必要があります。コントロール数の整数部分はループの繰り返し回数や間接指定用に使います。コントロール数の小数部分は整数部分の変更や終点用に使います。

\boxed{I} と \boxed{I} とキー操作の省略HP-11Cは \boxed{I} や \boxed{I} キーのときにはその前に \boxed{f} キーを押さなくてもよいように設計してあります。

\boxed{I} や \boxed{I} キーの前に \boxed{f} キーを押してもキーコードでは \boxed{f} を省略してしまいます。

Iレジスタの直接指定

Iレジスタの直接指定にはIレジスタ中の数値の操作も含まれます。

Iレジスタへのストアとリコール $\boxed{STO}\boxed{I}$ (store in R_i) と $\boxed{RCL}\boxed{I}$ (recall from R_i) は $R_0 \sim R_9$, $R_0 \sim R_9$ の普通の数値レジスタに \boxed{STO} や \boxed{RCL} するのと同じです。

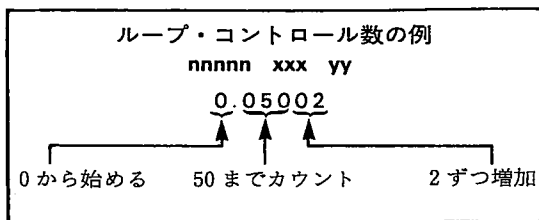
XとIレジスタの交換 $\boxed{f}\boxed{x\geq I}$ を押すと表示しているXレジスタ内の値とIレジスタ内の値を交換します。ちょうど $\boxed{x\geq y}$ を押すと表示しているXレジスタ内の値とYレジスタ内の値を交換するのと同じような働きです。

Iレジスタの値の増減 Iレジスタの数値を使ってループ計算の回数をコントロールするために **ISG** (increment, then skip if greater : 値を増加して、もし指定数より多かったら1ラインだけ飛び越す)と **DSE** (decrement, then skip if less than or equal : 値を減少して、もし指定数に等しいか少なかったら1ラインだけ飛び越す)があります。

ISG か **DSE** で一時的に計算を保留して、ループ・コントロール数が次の形式のどれにあたるかを比較します。

	±nnnnn	現在のカウンタの値
nnnnn.xxxyy , ここで	xxx	カウンタの最終値
	yy	1回ごとの増減値

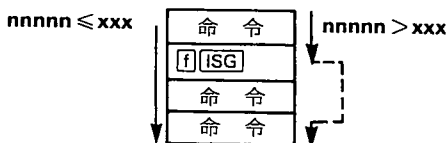
nnnnn はコントロール数の整数部分で、ループの繰り返し回数を数えてループ計算を続けるか終りにするかを判断したり、レジスタの間接指定、間接ジャンプ、間接サブルーチン・ジャンプ用に使います。この nnnnn は1~5桁の数を使いますが、何も指定しなければ0を入れたのと同じになります。**ISG** や **DSE** を1回実行するごとに nnnnn が増減します。



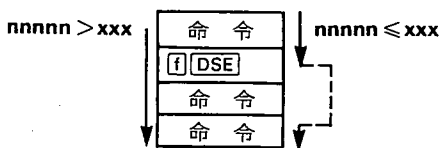
xxx はコントロール数の小数部分の初めの3桁です。**ISG** や **DSE** ごとに nnnnn が増減し、それから xxx と nnnnn を内部的に比較して (計算には無関係です) 増減した結果が最終値になったかどうか判定します。xxx は必ず3桁の数字になるようにします。例えば xxx に10を指定したいときは010と入力します。(xxx は最終値(目的値)なので **ISG** や **DSE** では変化しません。)

yy はコントロール数の小数部分で4～5桁目です。yy がコントロールの増減数で **ISG** や **DSE** ごとに nnnnn が yy ずつ増減します。yy は2桁の数字(例えば02, 03, 55)で指定します。yy を指定しないと自動的に01を指定したのと同じ働きをします。(yy は増減値なので **ISG** や **DSE** では変化しません。)

ISG と **DSE** の働き 前ページにあるようにプログラム走行中に **ISG** や **DSE** があると nnnnn が増減してから、第7章の条件判断と同じように nnnnn と xxx 比較します。**ISG** を通過するごとに nnnnn が yy ずつ増加しそれから nnnnn が xxx よりも大きいかどうか調べます。もし nnnnn の方が大きかったらプログラムの次の1ラインだけを飛びこしてその先を続けます。



DSE を通過するごとに nnnnn が yy ずつ減少しそれから nnnnn が xxx と等しい(または xxx よりも小さい)かどうか調べます。YESならプログラムの次の1ラインだけを飛びこしてその先を続けます。



R1の始めの値がこれのとき		ISG や DSE を繰り返し通ったときの nnnnn.xxx の値の変化				
回数 →	0	1	2	3	4	
ISG →	0.00602	2.00602	4.00602	6.00602	8.00602 (次を飛び越す)	
DSE →	6.00002	4.00002	2.00002	0.00002 (次を飛び越す)	-2.00002 (次を飛び越す)	

Iレジスタを使った間接指定

間接指定をしても Iレジスタの値が変化することはありません。逆に Iレジスタの値の整数部分の nnnnn でメモリー中の番号を探します。この間接指定はプログラム中に **ISG** や **DSE** と組み合わせてレジスタ番号やジャンプ先、サブルーチン・ジャンプ先を繰り返し指定するときなどに使います。

間接レジスタのストアとリコール **STO** (i) と **RCL** (i)* は Iレジスタの nnnnn の絶対値で指定したレジスタへのストアやリコールに使います。R₀ から R₉ は nnnnn=0 から 9 に対応し、R₀ から R₉ は nnnnn=10 から 19 に対応し、R₁ は nnnnn=20 に対応します。(次ページの間接指定の表を見てください。)

X と間接レジスタの交換 **f** **X \geq (i)** を押すと Xレジスタの値と nnnnn の絶対値で指定したレジスタの値が入れ替わります。

間接レジスタの直接四則演算 **STO** (+ , - , \times や \div のどれか) (i) は nnnnn の絶対値で指定したレジスタの直接四則演算に使います。

例

Iレジスタに 5.01202 をストアしてあると
nnnnn = 5 xxx = 012 yy = 02 なので

STO (i) = **STO** 5

f **X \geq (i)** = Xレジスタと R₅ の内容を入れ替え

STO + (i) = **STO** + 5

ラベルやライン番号の間接指定 プログラム走行中は

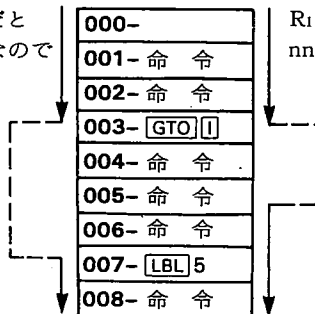
1. nnnnn \geq 0 であれば **GTO** (i) は nnnnn で指定したラベル (そこから探し始めて下方の一番始めの) ヘジャンプします。

* (i) と (i) キーの前の **f** キーは省略できます

2. $nnnnn < 0$ であれば **GTO I** は $nnnnn$ の絶対値と同じライン番号のところへジャンプします。

例えば

$R_1 = 5.010$ だと
 $nnnnn = 5$ なので



$R_1 = -5.010$ だと
 $nnnnn = -5$ なので

ラベルやライン番号の間接指定 プログラム走行中に **GSB I** があると **GTO I** と同じ方法で間接指定したラベルやライン番号にサブルーチン・ジャンプします。

間接指定の表

$nnnnn$	GTO または GSB I STO (i) や RCL (i) でジャンプする先	のときのレジスタ番号
0	f LBL 0	R_0
⋮	⋮	⋮
9	f LBL 9	R_9
10	" " A	$R_{.0}$
11	" " B	$R_{.1}$
12	" " C	$R_{.2}$
13	" " D	$R_{.3}$
14	" " E	$R_{.4}$
15	—	$R_{.5}$
16	—	$R_{.6}$
17	—	$R_{.7}$
18	—	$R_{.8}$
19	—	$R_{.9}$
20	—	R_1

ISG を使ったループ・コントロール

例 次のプログラムは ISG がどのように働くかを説明するためのものです。ループの途中で R_i の整数部分とその二乗をそれぞれ一時表示させているので ISG で R_i がどのように変化するかよくわかると思います。これは 2 から 50 までの偶数の二乗を順次求めるプログラムです。

キー操作

g **P/R**

f **CLEAR** **PRGM**

f **LBL** **A**

2

-

0

5

0

0

2

STO **I**

f **LBL** **1**

RCL **I**

g **INT**

f **PSE**

g **x²**

f **PSE**

f **ISG**

GTO **1**

g **RTN**

g **P/R**

表示

000-

PRGM モードにする。

000-

001-42,21,11

プログラムのラベル名

002-

2

カウンターの始の値

003-

48

(nnnnn)。

004-

0

005-

5

カウンターの判定値

006-

0

(最終値: xxx)

007-

0

008-

2

カウンターの増加値
(yy)。

009- 44 25

ループ・コントロール値を R_i にストア。

010-42,21, 1

ループの始め。

011- 45 25

R_i の値を呼び出す。

012- 43 44

その整数部分を取出す。

013- 42 31

整数を一時表示。

014- 43 11

その数の二乗。

015- 42 31

二乗値を一時表示。

016- 42 6

R_i を 2 だけ増加し、最終値 (50) よりも大きいかどうか調べる。もし最終値よりも大きかったら次の 1 ラインを飛びこす。

017- 22 1

ループの終りでラベル 1 へ戻る。

018- 43 32

プログラムの終り。
計算モードに戻す。

それでは **f****A** を押してプログラムをスタートさせてみましょう。

キー操作	表示	
f A	2.0000	計算をスタートさせると2を一時表示し、次にその二乗を一時表示する。数が2ずつ増加してはその二乗値を表示し、50に達するとその二乗を表示して停止する。
	4.0000	
	4.0000	
	16.0000	
	⋮	
	50.0000	
	2,500.0000	

それではどうしてこうなるのか説明しましょう。

- A** キーを押すと2.05002をループ・コントロール数として R_1 にストアします。

nnnnn	xxx	yy
(0000)2 そのときのカウンターの値	050 判定値	02 増加値

- ラベル1からのループで次のようになります。

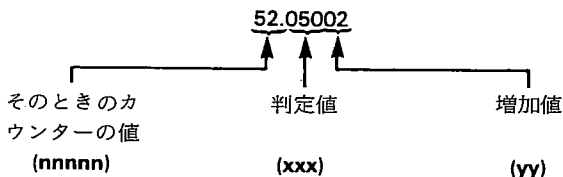
2と4(2の二乗)を一時表示したときには R_1 のカウンターの値は00002(nnnnn)ですが、次に増加値02(yy)だけ増加します。これで R_1 の新しい値は4.05002となり一たん計算を一時中止します。

nnnnn	xxx	yy
(0000)4 そのときのカウンターの値	050 判定値	02 増加値

新しいカウンターの値を判定値050(xxx)と比較します。カウンター値は判定値よりも小さいので、次のラインの命令の **GTO** 1でラベル1へ戻るといようにループ計算を続けます。

3. 偶数を25回表示(2~50)し、それぞれの二乗値を表示するとカウンター値は50よりも大きくなります。それで **[ISG]** (016ライン)の次の **[GTO]** 1 (017ライン)を飛びこして18ラインの **[RTN]**を実行するので、000ラインへ戻って停止します。

計算終了後に **[f]** **[FIX]** 5と押してから、**[RCL]** **[I]**を押してR₁の値を表示させると次のようになっています。



[f] **[FIX]** 4を押して **[FIX]** 4に戻しておきましょう。

ISGとDSEの限界 **[ISG]**や**[DSE]**によって2桁までならどんな値でも増加値や減少値を指定できます。しかしカウンターの値がnnnnnの5桁以内でなく6桁以上になるとこの増減値に影響がでできます。

例えば99,950.50055で**[ISG]**を通ると100,005.5006になります。始めの増加値は55でしたが、**[ISG]**を通ったために新しいR₁の値は100,005.50055になりますが全部で11桁になって表示できなくなり最後の桁で四捨五入して繰り上がります。そのため増加値yyは自動的に55でなく60に変化してしまいます。また999,945.5006のときに**[ISG]**を通ると1,000,005.5006になって四捨五入されて1,000,005.501になり増加値yyの部分が無くなってしまいます。そのためyyはそれまでの60から自動的に01に変わってしまいます。

R₁の直接指定

操 作 目 的	キー操作例	表 示
R ₁ にストア	12345 [STO] [I]	12345 12.345.0000

操 作 目 的	キー操作例	表 示
(表示をクリア)	\leftarrow	0.0000
R ₁ (先ほど $\text{STO } \boxed{1}$ でストアした値) をリコール	$\text{RCL } \boxed{1}$	12.345.0000
(表示をクリア)	\leftarrow	0.0000
XとIレジスタの交換 (先ほどのR ₁ がまだ残っています)	$\boxed{f} \boxed{x \geq 1}$	12.345.0000

R₁ を使った間接指定

操 作 目 的	キー操作例	表 示
(R ₁ に間接指定用つまりnnnnnとして3をストアする)	$3 \text{ STO } \boxed{1}$	3.0000
$\sqrt{7}$ を計算してR ₃ に間接ストア, リコール (R ₁ に3をストアしておいたのこのようになる)	$7 \sqrt{x}$ $\text{STO } \boxed{(i)}$ \leftarrow $\text{RCL } \boxed{(i)}$ \leftarrow	2.6458 2.6458 0.0000 2.6458 0.0000
間接指定を利用してXレジスタの0.0000とR ₃ の2.6458を交換する (R ₁ にnnnnnとして3が入っているから)	$\boxed{f} \boxed{x \geq (i)}$ $\boxed{f} \boxed{x \geq (i)}$	2.6458 0.0000

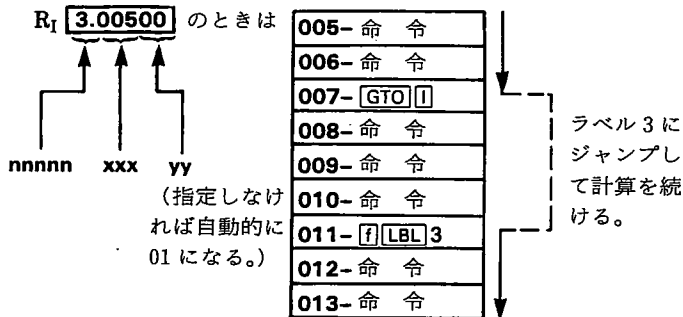
操 作 目 的	キー操作例	表 示
間接指定レジスタの直接 四則演算,ここでは R ₃ の 内容を π で割る	$f \pi$	3.1416
	STO \div (i)	3.1416
	RCL (i)	0.8422

レジスタの間接指定についてはこの本の第3部 [ISG] を使った [RCL] (i) (211 ページ) でも説明してあります。

プログラムの間接コントロール

ジャンプとサブルーチン・ジャンプのときのラベルの間接指定レジスタを間接指定したのと同じ方法でラベル番号やライン番号の間接指定もできます。131 ページに R_I の値とそのときの間接指定ラベルの表がありますので見てください。どのラベルも R_I の nnnnn が正数のときだけ指定できます。

プログラム走行中に [GTO] I に出会うと直ちにプログラム・メモリー
の下方へ nnnnn で指定したラベルを探してそこへジャンプします。
例えば I レジスタに 3.005 が入っていると

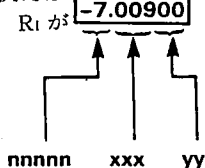


サブルーチン・ジャンプのラベル間接指定は [GTO] I が [GSB] I になる以外は上と同じです。

ジャンプとサブルーチン・ジャンプのときのライン番号の間接指定

ラベルの間接指定とほぼ同じ方法で、ジャンプとサブルーチン・ジャンプのときにライン番号の間接指定もできます。R_iのnnnnnが正数のときにはそれに対応するラベルへジャンプしますが、nnnnnが負数のときにはnnnnnの絶対値と同じライン番号のところへジャンプします。

例えば R_iが **-7.00900** のときは



(指定しなければ自動的に01になる。)

001- 命 令
002- 命 令
003- GTO I
004- 命 令
005- 命 令
006- 命 令
007- f LBL 3
008- 命 令
009- 命 令

007ラインへ
ジャンプして
計算を続行す
る。

第 3 部
各種のプログラム

第10章 各種プログラム

3 × 3 行列 (マトリクス) 計算

このプログラムは 3 × 3 の行列 A の行列式 $\det A$ と逆行列 C を求め (ただし行列式が 0 でないことが条件です), プログラムの一部を変更するだけでマトリクス形式の連立 1 次方程式の根も求められます。

計算式

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \quad B = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \quad X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

$$C = \frac{1}{\det A} \begin{bmatrix} \begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} & -\begin{vmatrix} a_{12} & a_{13} \\ a_{32} & a_{33} \end{vmatrix} & \begin{vmatrix} a_{12} & a_{13} \\ a_{22} & a_{23} \end{vmatrix} \\ -\begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{13} \\ a_{31} & a_{33} \end{vmatrix} & -\begin{vmatrix} a_{11} & a_{13} \\ a_{21} & a_{23} \end{vmatrix} \\ \begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix} & -\begin{vmatrix} a_{11} & a_{12} \\ a_{31} & a_{32} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} \end{bmatrix}$$

ただし C は A のマトリクスの逆行列,

$$\det A = a_{11}(a_{22}a_{33} - a_{23}a_{32}) - a_{12}(a_{21}a_{33} - a_{23}a_{31}) \\ + a_{13}(a_{21}a_{32} - a_{22}a_{31})$$

$$\begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad - bc.$$

$AX=B$ の形の連立方程式は $X=CB$ の形に変形して根を求めます。

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \frac{1}{\det A} \begin{bmatrix} b_1 \begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} - b_2 \begin{vmatrix} a_{12} & a_{13} \\ a_{32} & a_{33} \end{vmatrix} + b_3 \begin{vmatrix} a_{12} & a_{13} \\ a_{22} & a_{23} \end{vmatrix} \\ -b_1 \begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} + b_2 \begin{vmatrix} a_{11} & a_{13} \\ a_{31} & a_{33} \end{vmatrix} - b_3 \begin{vmatrix} a_{11} & a_{13} \\ a_{21} & a_{23} \end{vmatrix} \\ b_1 \begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix} - b_2 \begin{vmatrix} a_{11} & a_{12} \\ a_{31} & a_{32} \end{vmatrix} + b_3 \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} \end{bmatrix}$$

注1. マトリクス計算は掛算や割算を繰返すために数値の丸め（四捨五入）誤差が出る傾向があります。特殊の場合には正しく計算できないこともあります。

2. 連立方程式を幾つも解くときには、始めに行列式のサブルーチンでそれぞれの行列の行列式を求めておき、それから行列式のサブルーチンを連立方程式のサブルーチンに組替え、それぞれの行列式を R_0 にストアしてからその連立方程式を解くとよいでしょう。また149ページにも別のプログラムがあります。

3. 2×2 の行列は右のようにキーインしてください。
- $$\begin{bmatrix} a_{11} & a_{12} & 0 \\ a_{21} & a_{22} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

4. そのマトリクスが特異な場合は Error 0 を表示します。

主プログラム

キー操作	表示	キー操作	表示
f CLEAR PRGM	000-	RCL 1	027- 45 .1
f LBL 9	001-42,21, 9	RCL 0	028- 45 .0
f ISG	002- 42 6	RCL 8	029- 45 8
f PSE	003- 42 31	RCL 3	030- 45 .3
RCL (i)	004- 45 24	GSB 8	031- 32 8
x	005- 20	RCL 5	032- 45 5
x	006- 20	RCL 3	033- 45 .3
+	007- 40	RCL 1	034- 45 .1
STO 0	008- 44 0	RCL 7	035- 45 7
g RTN	009- 43 32	GSB 8	036- 32 8
f LBL B	010-42,21,12	RCL 8	037- 45 8
RCL 9	011- 45 9	RCL 7	038- 45 7
RCL 3	012- 45 .3	RCL 5	039- 45 5
RCL 2	013- 45 .2	RCL 0	040- 45 .0
RCL 0	014- 45 .0	GSB 8	041- 32 8
GSB 8	015- 32 8	GSB 2	042- 32 2
RCL 2	016- 45 .2	RCL 8	043- 45 8
RCL 7	017- 45 7	RCL 2	044- 45 .2
RCL 6	018- 45 6	RCL 1	045- 45 .1
RCL 3	019- 45 .3	RCL 9	046- 45 9
GSB 8	020- 32 8	GSB 8	047- 32 8
RCL 6	021- 45 6	RCL 1	048- 45 .1
RCL 0	022- 45 .0	RCL 6	049- 45 6
RCL 9	023- 45 9	RCL 5	050- 45 5
RCL 7	024- 45 7	RCL 2	051- 45 .2
GSB 8	025- 32 8	GSB 8	052- 32 8
GSB 2	026- 32 2	RCL 5	053- 45 5

キー操作	表示	キー操作	表示
$\boxed{\text{RCL}} 9$	054- 45 9	$\boxed{\text{R/S}}$	067- 31
$\boxed{\text{RCL}} 8$	055- 45 8	$\boxed{\text{g}} \boxed{\text{RTN}}$	068- 43 32
$\boxed{\text{RCL}} 6$	056- 45 6	$\boxed{\text{f}} \boxed{\text{LBL}} 2$	069-42,21, 2
$\boxed{\text{f}} \boxed{\text{LBL}} 8$	057-42,21, 8	$\boxed{\text{RCL}} 4$	070- 45 4
$\boxed{\times}$	058- 20	$\boxed{\text{g}} \boxed{\text{F7}} 0$	071-43, 6, 0
$\boxed{\text{R}} \downarrow$	059- 33	$\boxed{\text{R/S}}$	072- 31
$\boxed{\times}$	060- 20	$\boxed{\text{f}} \boxed{\text{LBL}} 3$	073-42,21, 3
$\boxed{\text{g}} \boxed{\text{R}} \uparrow$	061- 43 33	1	074- 1
$\boxed{-}$	062- 30	$\boxed{\text{STO}} \boxed{\text{I}}$	075- 44 25
$\boxed{\text{RCL}} 0$	063- 45 0	0	076- 0
$\boxed{\div}$	064- 10	$\boxed{\text{STO}} 4$	077- 44 4
$\boxed{\text{g}} \boxed{\text{F7}} 0$	065-43, 6, 0	$\boxed{\text{g}} \boxed{\text{RTN}}$	078- 43 32
$\boxed{\text{GTO}} 1$	066- 22 1		

レジスタ			R_i : 間接指定用
$R_0: \text{Det } A$	$R_1: b_1$	$R_2: b_2$	$R_3: b_3$
$R_4: x_1, x_2, x_3$	$R_5: a_{11}$	$R_6: a_{12}$	$R_7: a_{13}$
$R_8: a_{21}$	$R_9: a_{22}$	$R_{10}: a_{23}$	$R_{11}: a_{31}$
$R_{12}: a_{32}$	$R_{13}: a_{33}$		

行列式のサブルーチン

キー操作	表示	キー操作	表示
$\boxed{\text{f}} \boxed{\text{LBL}} \boxed{\text{A}}$	079-42,21,11	$\boxed{\text{RCL}} \boxed{\cdot} 0$	084- 45 .0
$\boxed{\text{g}} \boxed{\text{CF}} 0$	080-43, 5, 0	$\boxed{\text{RCL}} \boxed{\cdot} 2$	085- 45 .2
4	081- 4	$\boxed{\text{GSB}} 9$	086- 32 9
$\boxed{\text{STO}} \boxed{\text{I}}$	082- 44 25	$\boxed{\text{RCL}} 8$	087- 45 8
0	083- 0	$\boxed{\text{RCL}} \boxed{\cdot} 3$	088- 45 .3

キー操作	表示	キー操作	表示
GSB 9	089- 32 9	GSB 9	096- 32 9
RCL 9	090- 45 9	RCL 5	097- 45 5
RCL \square 1	091- 45 .1	RCL \square 3	098- 45 .3
GSB 9	092- 32 9	GSB 9	099- 32 9
CHS	093- 16	RCL 6	100- 45 6
RCL 7	094- 45 7	RCL \square 1	101- 45 .1
RCL \square 2	095- 45 .2	GSB 9	102- 32 9

連立方程式のサブルーチン

キー操作	表示	キー操作	表示
f LBL 1	079-42,21, 1	STO 3	088- 44 3
RCL (i)	080- 45 24	R \downarrow	089- 33
x	081- 20	STO 2	090- 44 2
STO + 4	082-44,40, 4	R \downarrow	091- 33
f ISG	083- 42 6	STO 1	092- 44 1
f PSE	084- 42 31	GSB 3	093- 32 3
g RTN	085- 43 32	GSB B	094- 32 12
f LBL C	086-42,21,13	g CF 0	095-43, 5, 0
g SF 0	087-43, 4, 0	RCL 4	096- 45 4

ステップ	操作説明	入力 数値/単位	キー操作	表示 数値/単位
1	主プログラムをキーイン (000-078 ライン)			
2	行列式のとき LBL Aのサブルーチン(079-102ラ イン)をキーイン		GTO \square 078 g P/R ... g P/R	

ステップ	操作説明	入力 数値/単位	キー操作	表示 数値/単位
3	USER モードにする			
4	行列の要素をストア			
		a_{11}	[STO] 5	
		a_{12}	[STO] 6	
		a_{13}	[STO] 7	
		a_{21}	[STO] 8	
		a_{22}	[STO] 9	
		a_{23}	[STO] 0	
		a_{31}	[STO] 1	
		a_{32}	[STO] 2	
		a_{33}	[STO] 3	
5	行列式を求める		[A]	Det A
6	逆行列のとき(任意操作)			
	逆行列の要素を求める		[B]	c_{11}
			[R/S]	c_{12}
			[R/S]	c_{13}
			[R/S]	c_{21}
			[R/S]	c_{22}
			[R/S]	c_{23}
			[R/S]	c_{31}
			[R/S]	c_{32}
			[R/S]	c_{33}
7	連立方程式のとき(任意操作)			
	079~102を削除		[G] [RTN] [G] [BST]	
			[G] [P/R]	102- 32 9
			[←]...[←]	078- 43 32
8	[LBL] 1のサブルーチン(079~096 ライン)をキーイン		[f] [LBL] 1 ... [G] [P/R]	079-42, 21, 1

ステップ	操作説明	入力 数値/単位	キー操作	表示 数値/単位
9	bの係数を入力して根を	b_1	ENTER	
	求める。	b_2	ENTER	
		b_3	C	x_1
			R/S	x_2
			R/S	x_3
10	別のbの係数のときは			
	ステップ9へ戻る。			
11	新しい行列式や逆行列			
	を求めるには079から		g RTN g BST	
	096ラインを削除。		g P/R	096- 45 4
			←...→	078- 43 32
	ステップ2へ戻る			

例1 $\begin{bmatrix} 14 & -8 \\ -8 & 12 \end{bmatrix}$ の逆行列を求めてから

$\begin{bmatrix} 14 & -8 \\ -8 & 12 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 20 \\ 5 \end{bmatrix}$ の根を求めてください。

キー操作 表示

USERモードにする。

f FIX 4

行列式のサブルーチンをキーイン。

14 STO 5

8 CHS STO 6

0 STO 7

8 CHS STO 8

12 STO 9

0 STO . 0

STO . 1 STO . 2

1 STO . 3

A

104.0000

Det. A

B

0.1154

C₁₁

キー操作

表示

R/S	0.0769	C_{12}
R/S	0.0000	C_{13}
R/S	0.0769	C_{21}
R/S	0.1346	C_{22}
R/S	0.0000	C_{23}
R/S	0.0000	C_{31}
R/S	0.0000	C_{32}
R/S	1.0000	C_{33}
g RTN g BST		
g P/R	102- 32 9	
← ← ... ←	078- 43 32	

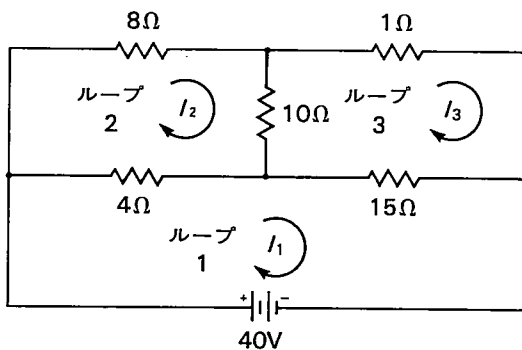
連立方程式のサブルーチンをキーイン。

キー操作

表示

g P/R		
20 ENTER 5 ENTER		
0 C	2.6923	x_1
R/S	2.2115	x_2
R/S	0.0000	x_3

例2 下図の各回路の電流を求めてください。



各回路の電流を連立方程式の形にまとめると

$$\begin{aligned}
 \text{ループ 1} & \quad 4I_1 - 4I_2 + 15I_3 - 40 = 0 \\
 \text{ループ 2} & \quad 4I_2 - 4I_1 + 8I_2 + 10I_2 - 10I_3 = 0 \\
 \text{ループ 3} & \quad 10I_3 - 10I_2 + 1I_3 + 15I_3 - 15I_1 = 0 \\
 \text{整理すると} & \quad 19I_1 - 4I_2 - 15I_3 = 40 \\
 & \quad -4I_1 + 22I_2 - 10I_3 = 0 \\
 & \quad -15I_1 - 10I_2 + 26I_3 = 0
 \end{aligned}$$

$$\begin{bmatrix} 19 & -4 & -15 \\ -4 & 22 & -10 \\ -15 & -10 & 26 \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \\ I_3 \end{bmatrix} = \begin{bmatrix} 40 \\ 0 \\ 0 \end{bmatrix} .$$

キー操作 表 示

19 **[STO]** 5
 4 **[CHS]** **[STO]** 6
 15 **[CHS]** **[STO]** 7
 4 **[CHS]** **[STO]** 8
 22 **[STO]** 9
 10 **[CHS]** **[STO]** **[.]** 0
 15 **[CHS]** **[STO]** **[.]** 1
 10 **[CHS]** **[STO]** **[.]** 2
 26 **[STO]** **[.]** 3

連立方程式のサブルーチンが入っているままのときはそれを削除して行列式のサブルーチンをキーイン。

[A] **2.402.0000**

行列式のサブルーチンを削除して連立方程式のサブルーチンをキーイン。

40 **[ENTER]**
 0 **[ENTER]** **[C]** **7.8601**
[R/S] **4.2298**
[R/S] **6.1615**

三元一次連立方程式

これはクラメルの法則を使って三元一次の連立方程式の根を求めるプログラムです。この前の 3×3 行列のプログラムよりも短く操作法も簡単になっています。

計算式 連立方程式は次のように表わすことができます。

$$AX = B.$$

$$\text{ここで, } A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

$$X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad B = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

$$\det A = a_{11}(a_{22}a_{33} - a_{23}a_{32}) - a_{12}(a_{21}a_{33} - a_{23}a_{31}) \\ + a_{13}(a_{21}a_{32} - a_{22}a_{31}).$$

x_i は次式で求めます

$$x_i = \frac{\det(i)}{\det A}$$

$\det A$ (行列 A の行列式) が0でなければ $\det(i)$ に列 B の要素を代入して x_i を求めます。

注 $\det A$ が0のときは三つの直線の交点がないためこのプログラムは使えません。このプログラムでは Error 0 を表示して計算を中止します。 $\det A$ が極めて0に近いときは、計算途中の桁落ちや $\det(i)/\det A$ の計算でエラーが発生することがあります。

キー操作	表示	キー操作	表示
\boxed{f} CLEAR \boxed{PRGM}	000-	\boxed{RCL} 5	029- 45 5
\boxed{f} \boxed{LBL} A	001-42,21,11	\boxed{RCL} 7	030- 45 7
1	002- 1	\boxed{GSB} 9	031- 32 9
$\boxed{\square}$	003- 48	\boxed{CHS}	032- 16
0	004- 0	\boxed{RCL} 3	033- 45 3
0	005- 0	\boxed{RCL} 8	034- 45 8
9	006- 9	\boxed{GSB} 9	035- 32 9
\boxed{STO} \boxed{I}	007- 44 25	\boxed{RCL} 1	036- 45 1
\boxed{f} \boxed{LBL} 2	008-42,21, 2	\boxed{RCL} 9	037- 45 9
\boxed{RCL} \boxed{I}	009- 45 25	\boxed{GSB} 9	038- 32 9
\boxed{g} \boxed{INT}	010- 43 44	\boxed{RCL} 2	039- 45 2
\boxed{f} \boxed{PSE}	011- 42 31	\boxed{RCL} 7	040- 45 7
\boxed{RCL} $\boxed{(i)}$	012- 45 24	\boxed{f} \boxed{LBL} 9	041-42,21, 9
$\boxed{R/S}$	013- 31	\boxed{f} \boxed{DSE}	042- 42 5
\boxed{STO} $\boxed{(i)}$	014- 44 24	\boxed{f} \boxed{PSE}	043- 42 31
\boxed{f} \boxed{ISG}	015- 42 6	\boxed{RCL} $\boxed{(i)}$	044- 45 24
\boxed{GTO} 2	016- 22 2	$\boxed{\times}$	045- 20
\boxed{GSB} 0	017- 32 0	$\boxed{\times}$	046- 20
\boxed{STO} 0	018- 44 0	$\boxed{+}$	047- 40
$\boxed{R/S}$	019- 31	\boxed{g} \boxed{RTN}	048- 43 32
\boxed{f} \boxed{LBL} 0	020-42,21, 0	\boxed{f} \boxed{LBL} \boxed{B}	049-42,21,12
0	021- 0	\boxed{STO} $\boxed{\cdot}$ 3	050- 44 .3
\boxed{STO} \boxed{I}	022- 44 25	$\boxed{R\downarrow}$	051- 33
\boxed{RCL} 6	023- 45 6	\boxed{STO} $\boxed{\cdot}$ 2	052- 44 .2
\boxed{RCL} 8	024- 45 8	$\boxed{R\downarrow}$	053- 33
\boxed{GSB} 9	025- 32 9	\boxed{STO} $\boxed{\cdot}$ 1	054- 44 .1
\boxed{RCL} 4	026- 45 4	1	055- 1
\boxed{RCL} 9	027- 45 9	\boxed{GSB} 7	056- 32 7
\boxed{GSB} 9	028- 32 9	$\boxed{R/S}$	057- 31

キー操作	表示	キー操作	表示
4	058- 4	$\boxed{\text{RCL}} \boxed{\cdot} 0$	075- 45 .0
$\boxed{\text{GSB}} 7$	059- 32 7	$\boxed{\text{g}} \boxed{\text{RTN}}$	076- 43 32
$\boxed{\text{R/S}}$	060- 31	$\boxed{\text{f}} \boxed{\text{LBL}} 8$	077-42,21, 8
7	061- 7	$\boxed{\text{RCL}} \boxed{\cdot} 1$	078- 45 .1
$\boxed{\text{GSB}} 7$	062- 32 7	$\boxed{\text{f}} \boxed{x \geq (i)}$	079- 42 23
$\boxed{\text{R/S}}$	063- 31	$\boxed{\text{STO}} \boxed{\cdot} 1$	080- 44 .1
$\boxed{\text{f}} \boxed{\text{LBL}} 7$	064-42,21, 7	$\boxed{\text{f}} \boxed{\text{ISG}}$	081- 42 6
$\boxed{\text{STO}} \boxed{\cdot} 4$	065- 44 .4	$\boxed{\text{f}} \boxed{\text{PSE}}$	082- 42 31
$\boxed{\text{STO}} \boxed{\text{I}}$	066- 44 25	$\boxed{\text{RCL}} \boxed{\cdot} 2$	083- 45 .2
$\boxed{\text{GSB}} 8$	067- 32 8	$\boxed{\text{f}} \boxed{x \geq (i)}$	084- 42 23
$\boxed{\text{GSB}} 0$	068- 32 0	$\boxed{\text{STO}} \boxed{\cdot} 2$	085- 44 .2
$\boxed{\text{RCL}} 0$	069- 45 0	$\boxed{\text{f}} \boxed{\text{ISG}}$	086- 42 6
$\boxed{+}$	070- 10	$\boxed{\text{f}} \boxed{\text{PSE}}$	087- 42 31
$\boxed{\text{STO}} \boxed{\cdot} 0$	071- 44 .0	$\boxed{\text{RCL}} \boxed{\cdot} 3$	088- 45 .3
$\boxed{\text{RCL}} \boxed{\cdot} 4$	072- 45 .4	$\boxed{\text{f}} \boxed{x \geq (i)}$	089- 42 23
$\boxed{\text{STO}} \boxed{\text{I}}$	073- 44 25	$\boxed{\text{STO}} \boxed{\cdot} 3$	090- 44 .3
$\boxed{\text{GSB}} 8$	074- 32 8	$\boxed{\text{g}} \boxed{\text{RTN}}$	091- 43 32

レジスター			R_i : 間接指定
R_0 : Det	R_1 : a_{11}	R_2 : a_{21}	R_3 : a_{31}
R_4 : a_{12}	R_5 : a_{22}	R_6 : a_{32}	R_7 : a_{13}
R_8 : a_{23}	R_9 : a_{33}	R_{10} : det(i)	R_{11} : b_1
R_{12} : b_2	R_{13} : b_3	R_{14} : 間接指定	

ステップ	操作説明	入力 数値/単位	キー操作	表示 数値/単位
1	プログラムをキーイン。			
2	USER モードにする。			

ステップ	操作説明	入力 数値/単位	キー操作	表示 数値/単位
3	3×3行列の係数をキーイン		A	1, R ₁
	(注 表示した数値をそのまま使う ときは R/S を押すだけでよい。)	a ₁₁	R/S	2, R ₂
		a ₂₁	R/S	3, R ₃
		a ₃₁	R/S	4, R ₄
		a ₁₂	R/S	5, R ₅
		a ₂₂	R/S	6, R ₆
		a ₃₂	R/S	7, R ₇
		a ₁₃	R/S	8, R ₈
		a ₂₃	R/S	9, R ₉
		a ₃₃	R/S	Det.
	行列の係数を見なおすには	A を押す		
4	列の係数をキーイン	b ₁	ENTER	b ₁
		b ₂	ENTER	b ₂
		b ₃	B	x ₁
			R/S	x ₂
			R/S	x ₃
5	列の係数を変えて計算するにはステップ4へ			
6	新しい3×3行列はステップ3へ			

例1 次の連立方程式を解いてください。

$$\begin{bmatrix} 19 & -4 & -15 \\ -4 & 22 & -10 \\ -15 & -10 & 26 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 40 \\ 0 \\ 0 \end{bmatrix}$$

キー操作

表示

f CLEAR **REG**

USER モードにする

f **FIX** 4

A

1.0000

0.0000

キー操作	表示		
19 R/S	2.0000	0.0000	
4 CHS R/S	3.0000	0.0000	
15 CHS R/S	4.0000	0.0000	
4 CHS R/S	5.0000	0.0000	
22 R/S	6.0000	0.0000	
10 CHS R/S	7.0000	0.0000	
15 CHS R/S	8.0000	0.0000	
10 CHS R/S	9.0000	0.0000	
26 R/S	2,402.0000		Det.
40 ENTER			
0 ENTER B	7.8601		x_1
R/S	4.2298		x_2
R/S	6.1615		x_3

例2 次の連立方程式を解いてください。

$$\begin{bmatrix} 19 & -4 & 4 \\ 5 & -12 & -10 \\ -15 & 8 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 5 \\ -3 \\ 4 \end{bmatrix}$$

キー操作	表示		
A	1.0000	19.0000	
R/S	2.0000	-4.0000	
5 R/S	3.0000	-15.0000	
R/S	4.0000	-4.0000	
R/S	5.0000	22.0000	
12 CHS R/S	6.0000	-10.0000	
8 R/S	7.0000	-15.0000	
4 R/S	8.0000	-10.0000	
R/S	9.0000	26.0000	
3 R/S	-264.0000		Det.
5 ENTER			
3 CHS ENTER			
4 B	-1.6667		x_1
R/S	-4.4091		x_2
R/S	4.7576		x_3

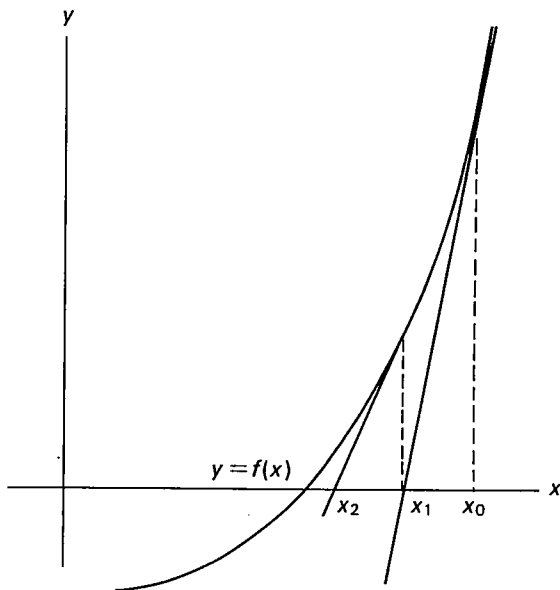
$f(x) = 0$ の根 (ニュートン法)

次式のような計算式の x (根) は普通の代数的方法で求めることはできません。

$$\ln x + 3x = 10.8074$$

それは x を左辺に集めても、 x について一つの項にまとめることができないからです。このようなときにこのニュートン法を使えば $f(x) = 0$ の根を求めることができます。ただし $f(x)$ を計算するプログラムだけは自分で作ってください。

この $f(x)$ を計算するプログラムは Xレジスタに x があるとしてスタートするようにしてください。主プログラムの長さは 68 ラインで、レジスタは $R_0 \sim R_4$ と R_1 を使っています。メモリーの残りは $f(x)$ の計算プログラム用に使えます。根を求める計算のときに根の近似値 x_0 をキーインする必要があります。この x_0 が実際の根に近いほど根 (答) が速く求まります。



このプログラムは次の2または3のどちらかと1が成立すると終了します。

1. 繰返し回数 \leq 繰返し限度
2. $|f(x)| \leq$ 誤差 (ϵ) の許容値
3. $|x$ の新しい値 $- x$ の古い値 $\leq \Delta x$ として決めた値

この ϵ と x はご自分でキーインしてもよいし、キーインしなければプログラムで自動的に決めるようにしてあります。

計算式

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$

このプログラムでは $f'(x)$ を求める代わりに次の近似式を使っています。

$$\text{ただし } \delta_0 = 10^{-5}x_0, \quad \delta_i = x_i - x_{i-1}$$

$$x_{i+1} = x_i - \delta_i \frac{f(x_i)}{f(x_i + \delta_i) - f(x_i)}$$

注

- 計算が終了すると $f(x) = 0$ の根 x を表示しますが、 R_2 にも入っています。
- 計算途中で **Error 3** を表示したら x_{i+1} の計算のときに0で割ったからです。このときにはエラーが発生したときにキーインした近似値 x_0 を少し変えて再スタートしてみてください。
- このプログラムは根を求めるときに繰返し計算の回数もチェックしています。020 から 022 ラインで R_1 に 50 を入れています。**Error 4** を表示したら繰返し回数が50よりも多くなったためです。そのときは021と020ラインを削除してから適当な数をキーインするとよいでしょう。
- このニュートン法のプログラムで x_{i+1} を毎回見られるようにすることもできます。それには047ライン（このプログラムのままで何の変更もしてなければ）の **[STO] 2** の後に **[f] [PSE]** をキーインするだけです。
- 許容誤差 ϵ を小さくしすぎると、計算途中の桁落ちのために正しい答を求められないことがあります。

キー操作	表示	キー操作	表示
f CLEAR PRGM	000-	GSB C	029- 32 13
f LBL A	001-42,21,11	STO 2	030- 44 2
EEX	002- 26	RCL 0	031- 45 0
CHS	003- 16	RCL 1	032- 45 1
5	004- 5	+	033- 40
STO 0	005- 44 0	GSB C	034- 32 13
EEX	006- 26	RCL 2	035- 45 2
CHS	007- 16	-	036- 30
8	008- 8	g x=0	037- 43 40
STO 3	009- 44 3	STO .9	038- 44 .9
STO 4	010- 44 4	RCL 2	039- 45 2
R/S	011- 31	$x \geq y$	040- 34
STO 3	012- 44 3	÷	041- 10
R/S	013- 31	RCL 0	042- 45 0
STO 4	014- 44 4	x	043- 20
g CLR	015- 43 35	CHS	044- 16
g RTN	016- 43 32	RCL 1	045- 45 1
f LBL B	017-42,21,12	+	046- 40
STO 2	018- 44 2	STO 2	047- 44 2
STO x 0	019-44,20, 0	g LSTx	048- 43 36
5	020- 5	-	049- 30
0	021- 0	STO 0	050- 44 0
STO I	022- 44 25	f DSE	051- 42 5
GSB 1	023- 32 1	GTO 2	052- 22 2
RCL 2	024- 45 2	GSB 9	053- 32 9
g RTN	025- 43 32	f LBL 2	054-42,21, 2
f LBL 1	026-42,21, 1	RCL 2	055- 45 2
RCL 2	027- 45 2	GSB C	056- 32 13
STO 1	028- 44 1	g ABS	057- 43 16

キー操作	表示	キー操作	表示
$\boxed{\text{RCL}} 3$	058- 45 3	$\boxed{\text{g}} \boxed{\text{ABS}}$	064- 43 16
$\boxed{x} \geq \boxed{y}$	059- 34	$\boxed{\text{f}} \boxed{x} \leq \boxed{y}$	065- 42 10
$\boxed{\text{f}} \boxed{x} > \boxed{y}$	060- 42 20	$\boxed{\text{g}} \boxed{\text{RTN}}$	066- 43 32
$\boxed{\text{GTO}} 1$	061- 22 1	$\boxed{\text{GTO}} 1$	067- 22 1
$\boxed{\text{RCL}} 4$	062- 45 4	$\boxed{\text{f}} \boxed{\text{LBL}} \boxed{\text{C}}$	068-42,21,13
$\boxed{\text{RCL}} 0$	063- 45 0		

	レジスター		R_1 : 繰返し回数
$R_0: \delta_i$	R_1 : 一時使用	R_2 : 一時使用	$R_3: \epsilon$
$R_4: \Delta x$	R_5 — R_8 : 未使用		

ステップ	内容説明	入力 数値/単位	キー操作	表示 数値/単位
1	プログラムをキーイン			
2	$\boxed{\text{LBL}} \boxed{\text{C}}$ にジャンプ		$\boxed{\text{GTO}} \boxed{\text{C}}$	
3	PRGM モードにする		$\boxed{\text{g}} \boxed{\text{P/R}}$	068-42,21,13
4	$f(x)$ を計算するプログラムをキーイン			
5	計算モードに戻す		$\boxed{\text{g}} \boxed{\text{P/R}}$	
6	USER モードにする			
7	許容誤差などの指定		$\boxed{\text{A}}$	10^{-8}
8	任意操作 ϵ の入力	ϵ	$\boxed{\text{R/S}}$	ϵ
	ϵ は 10^{-8} のままで Δx を 変えたいときは右の操作			
	をしてステップ9へ		$\boxed{\text{R/S}}$	10^{-8}
9	任意操作 Δx の入力	Δx	$\boxed{\text{R/S}}$	Δx
10	初期値を入れて根の計算			
		x_0	$\boxed{\text{B}}$	x_i

ステップ	内容説明	入力 数値/単位	キー操作	表示 数値/単位
11	別の初期値で計算する			
	ときはステップ10へ			
12	別の許容誤差にすときはステップ8へ			
13	別の $f(x)$ の根を求めるときはプロ			
	グラムの終りへジャンプ		g RTN g BST	
	PRGMモードにして削除		g P/R	
	しながら LBL C まで戻る		← ... ←	068-42,21,13
	それからステップ4へ			

例 $x_0 = 2$ として次式の根を求めてください

$$f(x) = x^6 - x - 1 = 0$$

キー操作

表示

GTO **C**

g **P/R**

068-42,21,13

STO 5

6 **↵**

RCL 5 **□** 1 **□**

075- 30

g **P/R**

f **FIX** 9

USER モードにする

A

0.00000010 e と Δx はこのまま使用

2 **B**

1.134724138 x_i

C

-0.000000004 $f(x_i)$

不連続点の数値積分

これは関数の計算式がわかっていなくても、区間内を有限個で等間隔に区分したときの関数値がわかっているときの積分値を求めるプログラムです。ここでは台形法またはシンプソン法で積分の近似値を求めます。

計算式

積分区間を n 等分すると x_0, x_1, \dots, x_n ($x_j = x_0 + jh$, $j = 0, 1, 2, \dots, n$) になり、そのときの関数値 $f(x_0), f(x_1), \dots, f(x_n)$ がわかっているものとします。

積分 $\int_{x_0}^{x_n} f(x) dx$ の近似値は次の式で計算できます。

1. 台形法

$$\int_{x_0}^{x_n} f(x) dx \sim \frac{h}{2} \left[f(x_0) + 2 \left(\sum_{j=1}^{n-1} f(x_j) \right) + f(x_n) \right]$$

2. シンプソン法

$$\int_{x_0}^{x_n} f(x) dx \sim \frac{h}{3} [f(x_0) + 4f(x_1) + 2f(x_2) + \dots + 4f(x_{n-3}) + 2f(x_{n-2}) + 4f(x_{n-1}) + f(x_n)]$$

シンプソン法のときには n が偶数であることが条件になります。 n が奇数のときには Error 0 の表示になります。

キー操作	表示	キー操作	表示
\boxed{f} CLEAR \boxed{PRGM}	000-	\boxed{STO} 0	005- 44 0
\boxed{f} \boxed{LBL} \boxed{A}	001-42,21,11	\boxed{STO} 5	006- 44 5
\boxed{STO} 4	002- 44 4	0	007- 0
$\boxed{R/S}$	003- 31	\boxed{STO} 3	008- 44 3
\boxed{f} \boxed{LBL} \boxed{B}	004-42,21,12	\boxed{f} \boxed{LBL} 1	009-42,21, 1

キー操作	表示	キー操作	表示
$\boxed{R/S}$	010- 31	$\boxed{GSB} 3$	035- 32 3
$\boxed{f} \boxed{LBL} \boxed{B}$	011-42,21,12	3	036- 3
$\boxed{STO} 1$	012- 44 1	$\boxed{RCL} 5$	037- 45 5
$\boxed{GSB} 2$	013- 32 2	$\boxed{f} \boxed{LBL} 0$	038-42,21, 0
\boxed{ENTER}	014- 36	$\boxed{RCL} 1$	039- 45 1
$\boxed{+}$	015- 40	$\boxed{-}$	040- 30
$\boxed{STO} \boxed{+} 5$	016-44,40, 5	$\boxed{RCL} 4$	041- 45 4
1	017- 1	$\boxed{\times}$	042- 20
$\boxed{STO} \boxed{+} 3$	018-44,40, 3	$\boxed{x \geq y}$	043- 34
$\boxed{RCL} 3$	019- 45 3	$\boxed{\div}$	044- 10
$\boxed{R/S}$	020- 31	$\boxed{R/S}$	045- 31
$\boxed{f} \boxed{LBL} \boxed{B}$	021-42,21,12	$\boxed{f} \boxed{LBL} 2$	046-42,21, 2
$\boxed{STO} 1$	022- 44 1	\boxed{ENTER}	047- 36
$\boxed{GSB} 2$	023- 32 2	$\boxed{+}$	048- 40
$\boxed{STO} \boxed{+} 5$	024-44,40, 5	$\boxed{STO} \boxed{+} 0$	049-44,40, 0
1	025- 1	$\boxed{g} \boxed{RTN}$	050- 43 32
$\boxed{STO} \boxed{+} 3$	026-44,40, 3	$\boxed{f} \boxed{LBL} 3$	051-42,21, 3
$\boxed{RCL} 3$	027- 45 3	2	052- 2
$\boxed{GTO} 1$	028- 22 1	$\boxed{\pm}$	053- 10
$\boxed{f} \boxed{LBL} \boxed{C}$	029-42,21,13	$\boxed{f} \boxed{FRAC}$	054- 42 44
2	030- 2	$\boxed{g} \boxed{x=0}$	055- 43 40
$\boxed{RCL} 0$	031- 45 0	$\boxed{g} \boxed{RTN}$	056- 43 32
$\boxed{GTO} 0$	032- 22 0	0	057- 0
$\boxed{f} \boxed{LBL} \boxed{D}$	033-42,21,14	$\boxed{\div}$	058- 10
$\boxed{RCL} 3$	034- 45 3	$\boxed{g} \boxed{RTN}$	059- 43 32

レジスタ			R_1 : 未使用
R_0 : 使用	$R_1: f(x_i)$	R_2 : 未使用	$R_3: i$
$R_4: h$	R_5 : 使用	R_6-R_9 : 未使用	

ステップ	内容説明	入力 数値/単位	キー操作	表示 数値/単位
1	プログラムをキーイン			
2	USER モードにする			
3	x の値の増加分を入力	h	[A]	h
4	x_j (ただし $j=0, 1, \dots, n$) のときの関数値を入力	$f(x_j)$	[B]	j
5	台形法で積分値を求める			
	または		[C]	Trap f
	シンプソン法で求める (n は偶数に限る)		[D]	Simp f
6	新しい計算はステップ3へ			

例 $f(x_j)$ (ただし $j=0, 1, \dots, 8$) の値が下表のときに次の積分の近似値を求めてください

$$\int_0^2 f(x) dx$$

台形法とシンプソン法の両方でやってください。この場合 h は 0.25 です。

i	0	1	2	3	4	5	6	7	8
x_j	0	0.25	0.5	0.75	1	1.25	1.5	1.75	2
$f(x_j)$	2	2.8	3.8	5.2	7	9.2	12.1	15.6	20

キー操作 表示

[f][FIX]2

USER モードにする

.25 [A] 0.25

2 [B] 0.00

2.8 [B] 1.00

3.8 [B] 2.00

5.2 [B] 3.00

キー操作

表示

7 B	4.00	
9.2 B	5.00	
12.1 B	6.00	
15.6 B	7.00	
20 B	8.00	
C	16.68	台形法
D	16.56	シンプソン法

曲線のあてはめ

HP-11Cには **L.R.** (linear regression: 直線のあてはめ) という強力な内蔵関数があって、沢山のデータにあてはまる直線の係数を簡単に素早く求めることができます。

このプログラムでは次の3種の曲線のどれかを指定して係数 a 、 b を求められます。

1. 指数曲線 $y = ae^{bx}$ ($a > 0$)
2. 対数曲線 $y = a + b \ln x$
3. 累乗 (べき乗) 曲線 $y = ax^b$ ($a > 0$)

どれもデータを $Y = A + bX$ の形に変形して計算します。

係数 A 、 b は次の連立方程式を解いて求めます。

$$\begin{bmatrix} n & \sum X_i \\ \sum X_i & \sum X_i^2 \end{bmatrix} \begin{bmatrix} A \\ b \end{bmatrix} = \begin{bmatrix} \sum Y_i \\ \sum (Y_i X_i) \end{bmatrix}$$

n はデータ数です。

この式の変数は次の通りです

曲線の種類	A	X_i	Y_i	コード
指数	$\ln a$	x_i	$\ln y_i$	1
対数	a	$\ln x_i$	y_i	2
累乗	$\ln a$	$\ln x_i$	$\ln y_i$	3

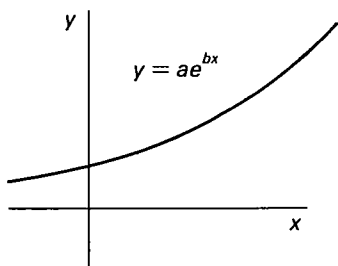
決定係数

$$r^2 = \frac{A \sum Y_i + b \sum X_i Y_i - \frac{1}{n} (\sum Y_i)^2}{\sum (Y_i^2) - \frac{1}{n} (\sum Y_i)^2}$$

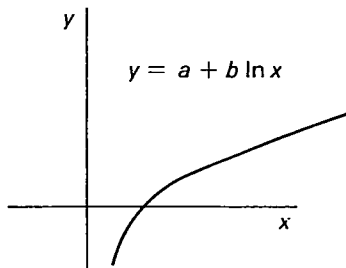
x と y の値を入力する前にコード番号を入力してどの曲線をあてはめるか指定します。

決定係数はデータにその曲線がどの程度うまくあてはまるかを示すものです。 r^2 が 1.00 に近いものは 0 に近いときよりもよくあてはまっています。係数 a 、 b は左ページで説明した通りです。

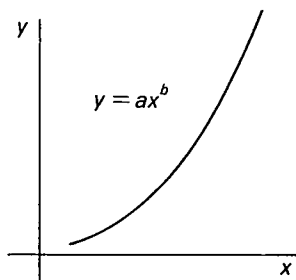
指数曲線のあてはめ
コード=1



対数曲線のあてはめ
コード=2



累乗曲線のあてはめ
コード=3



- 注 ● このプログラムはデータを162ページのように変形してから最小二乗法で解いています。
- 指数曲線のときに y_i に負数や0があるとエラーになります。対数曲線のあてはめのために x_i に負数や0があるとエラーになります。累乗曲線のあてはめのために x_i と y_i のどちらも正数（0は含みません）でないとエラーになります。
 - x か y のどちらかまたは両方とも最大と最小の差が小さいときには係数の計算精度が悪くなります。
 - プログラム計算途中で数値レジスタ全数をクリアしますので、計算前にレジスタに入れておいた値は消えてしまいます。

キー操作	表示	キー操作	表示
\boxed{f} CLEAR \boxed{PRGM}	000-	6	006- 6
\boxed{f} \boxed{LBL} \boxed{A}	001-42,21,11	$\boxed{+}$	007- 40
\boxed{f} CLEAR \boxed{REG}	002- 42 34	\boxed{STO} \boxed{I}	008- 44 25
\boxed{f} \boxed{FIX} 2	003-42, 7, 2	\boxed{R} $\boxed{\downarrow}$	009- 33
\boxed{ENTER}	004- 36	\boxed{g} \boxed{SF} 0	010-43, 4, 0
\boxed{ENTER}	005- 36	\boxed{g} \boxed{SF} 1	011-43, 4, 1

キー操作	表示	キー操作	表示
$\boxed{\text{GTO}} \boxed{1}$	012- 22 25	$\boxed{\text{g}} \boxed{\Sigma^-}$	036- 43 49
$\boxed{\text{f}} \boxed{\text{LBL}} \boxed{7}$	013-42,21, 7	$\boxed{\text{GTO}} \boxed{9}$	037- 22 9
$\boxed{\text{g}} \boxed{\text{CF}} \boxed{1}$	014-43, 5, 1	$\boxed{\text{f}} \boxed{\text{LBL}} \boxed{\text{B}}$	038-42,21,12
$\boxed{\text{GTO}} \boxed{9}$	015- 22 9	$\boxed{\text{f}} \boxed{\text{L.R.}}$	039- 42 49
$\boxed{\text{f}} \boxed{\text{LBL}} \boxed{8}$	016-42,21, 8	$\boxed{\text{g}} \boxed{\text{F?}} \boxed{0}$	040-43, 6, 0
$\boxed{\text{g}} \boxed{\text{CF}} \boxed{0}$	017-43, 5, 0	$\boxed{e^x}$	041- 12
$\boxed{\text{f}} \boxed{\text{LBL}} \boxed{9}$	018-42,21, 9	$\boxed{\text{R/S}}$	042- 31
$\boxed{\text{R/S}}$	019- 31	$\boxed{x \geq y}$	043- 34
$\boxed{\text{g}} \boxed{\text{F?}} \boxed{0}$	020-43, 6, 0	$\boxed{\text{R/S}}$	044- 31
$\boxed{\text{g}} \boxed{\text{LN}}$	021- 43 12	$\boxed{\text{f}} \boxed{y.r}$	045- 42 48
$\boxed{x \geq y}$	022- 34	$\boxed{x \geq y}$	046- 34
$\boxed{\text{g}} \boxed{\text{F?}} \boxed{1}$	023-43, 6, 1	$\boxed{\text{g}} \boxed{x^2}$	047- 43 11
$\boxed{\text{g}} \boxed{\text{LN}}$	024- 43 12	$\boxed{\text{g}} \boxed{\text{RTN}}$	048- 43 32
$\boxed{\text{RCL}} \boxed{6}$	025- 45 6	$\boxed{\text{f}} \boxed{\text{LBL}} \boxed{\text{D}}$	049-42,21,14
$\boxed{\text{g}} \boxed{x \neq 0}$	026- 43 30	1	050- 1
$\boxed{\text{GTO}} \boxed{6}$	027- 22 6	$\boxed{\text{STO}} \boxed{6}$	051- 44 6
$\boxed{\text{R}\downarrow}$	028- 33	$\boxed{\text{GTO}} \boxed{9}$	052- 22 9
$\boxed{\Sigma^+}$	029- 49	$\boxed{\text{f}} \boxed{\text{LBL}} \boxed{\text{C}}$	053-42,21,13
$\boxed{\text{GTO}} \boxed{9}$	030- 22 9	$\boxed{\text{g}} \boxed{\text{F?}} \boxed{1}$	054-43, 6, 1
$\boxed{\text{f}} \boxed{\text{LBL}} \boxed{6}$	031-42,21, 6	$\boxed{\text{g}} \boxed{\text{LN}}$	055- 43 12
$\boxed{\text{R}\downarrow}$	032- 33	$\boxed{\text{f}} \boxed{y.r}$	056- 42 48
0	033- 0	$\boxed{\text{g}} \boxed{\text{F?}} \boxed{0}$	057-43, 6, 0
$\boxed{\text{STO}} \boxed{6}$	034- 44 6	$\boxed{e^x}$	058- 12
$\boxed{\text{R}\downarrow}$	035- 33		

レジスタ			R_i コード
$R_0: i$	$R_1: \Sigma x_i$	$R_2: \Sigma x_i^2$	$R_3: \Sigma y_i$
$R_4: \Sigma y_i^2$	$R_5: \Sigma x_i y_i$	$R_6: 1$ または 0	R_7-R_9 : 未使用

ステップ	内容説明	入力 数値/単位	キー操作	表示 数値/単位
1	プログラムをキーイン			
2	USER モードにする			
3	曲線の種類を選ぶ			
	指数曲線	1	[A]	1.00
	対数曲線	2	[A]	2.00
	累乗曲線	3	[A]	3.00
4	x_i と y_i の入力	x_i	[ENTER]	
	(全部のデータについて ステップ4を繰返す)	y_i	[R/S]	i
5	係数の計算		[B]	a
			[R/S]	b
			[R/S]	r^2
6	x のときの y の推定値			
	\hat{y} の計算	x	[C]	\hat{y}
7	入れ間違えたデータの取消し			
	ステップ4でデータを入れ間違えた		[D]	
	ときは [D] を押してから入れ間違	$x_{j, err}$	[ENTER]	
	えたデータをもう一度入力する。	$y_{j, err}$	[R/S]	$i-1$
	それからステップ4に戻って正しい			
	データを入れ直す			

例1

(指数曲線, コード = 1)

x_i	0.72	1.31	1.95	2.58	3.14
y_i	2.16	1.61	1.16	0.85	0.5

答

$$a = 3.45, \quad b = -0.58$$

$$y = 3.45 e^{-0.58x}$$

$$r^2 = 0.98$$

キー操作 表 示

USER モードにする

1 [A] 1.00

.72 [ENTER] 2.16 [R/S]

1.31 [ENTER] 1.61 [R/S]

1.95 [ENTER] 1.16 [R/S]

2.58 [ENTER] .85 [R/S]

3.14 [ENTER] .5 [R/S]

[B] 3.45 a [R/S] -0.58 b [R/S] 0.98 r^2 1.5 [C] 1.44 \hat{y}

例 2

(対数曲線, コード = 2)

x_i	3	4	6	10	12
y_i	1.5	9.3	23.4	45.8	60.1

答

$$a = -47.02, \quad b = 41.39$$

$$y = 47.02 + 41.39 \ln x$$

$$r^2 = 0.98$$

$$x = 8 \text{ のとき } \hat{y} = 39.06$$

$$x = 14.5 \text{ のとき } \hat{y} = 63.67$$

キー操作 表 示

2 [A] 2.00

3 [ENTER] 1.5 [R/S]

4 [ENTER] 9.3 [R/S]

6 [ENTER] 23.4 [R/S]

キー操作 表示

10	ENTER	45.8	R/S		
12	ENTER	60.1	R/S		
B				-47.02	a
R/S				41.39	b
R/S				0.98	r^2
8	C			39.06	\hat{y}
14.5	C			63.67	\hat{y}

例 3

(累乗曲線, コード=3)

x_i	1.3	2.6	4	5.4	7.1
Y_i	3.74	6.15	7.21	8.03	8.84

答

$$a = 3.49, b = 0.50$$

$$y = 3.49x^{0.50}$$

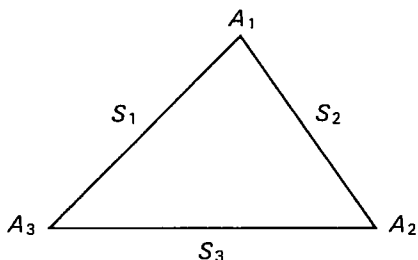
$$r^2 = 0.97$$

キー操作 表示

3	A			3.00	
1.3	ENTER	3.74	R/S		
2.6	ENTER	6.15	R/S		
4	ENTER	7.21	R/S		
5.4	ENTER	8.03	R/S		
7.1	ENTER	8.84	R/S		
B				3.49	a
R/S				0.50	b
R/S				0.97	r^2

三角形の計算

このプログラムは平面上の三角形の辺の長さ、角度と面積の計算に使えます。



上図の三角形の6個のパラメータ（三つの辺と三つの角）の内どれか三つがわかれば三角形が決まります。（ただし三つの角のときだけは決められません。）つまり次の五つのどれかであればこのプログラムで計算できます。二つの辺とその間の角（二辺挟角;SAS）、二つの角とその間の辺（二角挟辺;ASA）、二辺とその隣の角（二辺隣角;SSA—答が2組のことがあります）、二つの角とその隣の辺（二角隣辺;AAS）、三つの辺（三辺既知;SSS）の5通りです。

三角形の三つのデータは時計回りの順に入力し、答の表示も時計回りの順になります。

- 注
- 角度は DEG, RAD, GRAD のどれでも使えます。計算開始前に必要な角度単位に指定してください。
 - このプログラムの辺と角の表記法は普通の表記法とは違うので注意してください。（つまり A_1 の対辺は S_1 ではありません。）
 - 角度は10進法で入力してください。角度が度・分・秒のときは $\boxed{g} \boxed{\rightarrow H}$ で10進法に変換する必要があります。
 - 三つの角のうちどれかが0に近いときは計算精度が悪くなります。

キー操作	表示	キー操作	表示
\boxed{f} CLEAR \boxed{PRGM}	000-	\boxed{RCL} 2	029- 45 2
\boxed{f} \boxed{LBL} A	001-42,21,11	\boxed{RCL} 6	030- 45 6
\boxed{STO} 5	002- 44 5	$\boxed{+}$	031- 40
$\boxed{R}\downarrow$	003- 33	\boxed{SIN}	032- 23
\boxed{STO} 3	004- 44 3	$\boxed{\div}$	033- 10
$\boxed{R}\downarrow$	005- 33	\boxed{RCL} 1	034- 45 1
\boxed{STO} 1	006- 44 1	$\boxed{\times}$	035- 20
\boxed{RCL} 3	007- 45 3	\boxed{STO} 3	036- 44 3
\boxed{g} $\boxed{\rightarrow P}$	008- 43 26	\boxed{GTO} 0	037- 22 0
\boxed{g} $\boxed{x^2}$	009- 43 11	\boxed{f} \boxed{LBL} C	038-42,21,13
\boxed{RCL} 5	010- 45 5	\boxed{STO} 4	039- 44 4
\boxed{g} $\boxed{x^2}$	011- 43 11	$\boxed{R}\downarrow$	040- 33
$\boxed{-}$	012- 30	\boxed{STO} 2	041- 44 2
\boxed{RCL} 1	013- 45 1	$\boxed{R}\downarrow$	042- 33
\boxed{RCL} 3	014- 45 3	\boxed{STO} 1	043- 44 1
$\boxed{\times}$	015- 20	\boxed{RCL} 4	044- 45 4
2	016- 2	\boxed{RCL} 2	045- 45 2
$\boxed{\times}$	017- 20	$\boxed{+}$	046- 40
$\boxed{\div}$	018- 10	\boxed{SIN}	047- 23
\boxed{g} \boxed{COS}	019- 43 24	\boxed{RCL} 4	048- 45 4
\boxed{STO} 2	020- 44 2	\boxed{SIN}	049- 23
\boxed{GTO} 0	021- 22 0	$\boxed{\div}$	050- 10
\boxed{f} \boxed{LBL} B	022-42,21,12	\boxed{RCL} 1	051- 45 1
\boxed{STO} 2	023- 44 2	$\boxed{\times}$	052- 20
$\boxed{R}\downarrow$	024- 33	\boxed{STO} 3	053- 44 3
\boxed{STO} 1	025- 44 1	\boxed{GTO} 0	054- 22 0
$\boxed{R}\downarrow$	026- 33	\boxed{f} \boxed{LBL} E	055-42,21,15
\boxed{STO} 6	027- 44 6	\boxed{STO} 4	056- 44 4
\boxed{SIN}	028- 23	$\boxed{R}\downarrow$	057- 33

キー操作	表示	キー操作	表示
$\boxed{\text{STO}} 3$	058- 44 3	$f \boxed{\text{LBL}} \boxed{\text{D}}$	087-42,21,14
$\boxed{\text{R}} \downarrow$	059- 33	$\boxed{\text{STO}} 3$	088- 44 3
$\boxed{\text{STO}} 1$	060- 44 1	$\boxed{\text{R}} \downarrow$	089- 33
$\boxed{\text{RCL}} 3$	061- 45 3	$\boxed{\text{STO}} 2$	090- 44 2
$\boxed{\text{RCL}} 4$	062- 45 4	$\boxed{\text{R}} \downarrow$	091- 33
$\boxed{\text{SIN}}$	063- 23	$\boxed{\text{STO}} 1$	092- 44 1
$\boxed{\text{RCL}} 1$	064- 45 1	$f \boxed{\text{LBL}} 0$	093-42,21, 0
$\boxed{+}$	065- 10	$\boxed{\text{RCL}} 2$	094- 45 2
$\boxed{\times}$	066- 20	$\boxed{\text{RCL}} 1$	095- 45 1
$\boxed{g} \boxed{\text{SIN}}^{\circ}$	067- 43 23	$f \boxed{\rightarrow} \boxed{\text{R}}$	096- 42 26
$\boxed{\text{RCL}} 4$	068- 45 4	$\boxed{\text{RCL}} 3$	097- 45 3
$\boxed{+}$	069- 40	$\boxed{x \geq y}$	098- 34
$\boxed{\text{GSB}} 9$	070- 32 9	$\boxed{-}$	099- 30
$\boxed{\text{STO}} 2$	071- 44 2	$\boxed{g} \boxed{\rightarrow} \boxed{\text{P}}$	100- 43 26
$\boxed{\text{GSB}} 0$	072- 32 0	$\boxed{\text{STO}} 5$	101- 44 5
$\boxed{\text{RCL}} 1$	073- 45 1	$\boxed{x \geq y}$	102- 34
$\boxed{\text{RCL}} 3$	074- 45 3	$\boxed{\text{STO}} 4$	103- 44 4
$f \boxed{x \leq y}$	075- 42 10	$\boxed{\text{RCL}} 2$	104- 45 2
$\boxed{\text{GTO}} 8$	076- 22 8	$\boxed{+}$	105- 40
2	077- 2	$\boxed{\text{GSB}} 9$	106- 32 9
$\boxed{\text{R/S}}$	078- 31	$\boxed{\text{STO}} 6$	107- 44 6
$\boxed{\text{RCL}} 6$	079- 45 6	$\boxed{\text{SIN}}$	108- 23
$\boxed{\text{GSB}} 9$	080- 32 9	$\boxed{\times}$	109- 20
$\boxed{\text{STO}} 6$	081- 44 6	$\boxed{\text{RCL}} 1$	110- 45 1
$\boxed{\text{RCL}} 4$	082- 45 4	$\boxed{\times}$	111- 20
$\boxed{+}$	083- 40	2	112- 2
$\boxed{\text{GSB}} 9$	084- 32 9	$\boxed{+}$	113- 10
$\boxed{\text{STO}} 2$	085- 44 2	$\boxed{\text{STO}} 0$	114- 44 0
$\boxed{\text{GTO}} 0$	086- 22 0	$\boxed{\cdot}$	115- 48

キー操作	表示	キー操作	表示
0	116- 0	$\boxed{\text{GTO}} 2$	124- 22 2
0	117- 0	$\boxed{f} \boxed{\text{LBL}} 8$	125-42,21, 8
6	118- 6	$\boxed{g} \boxed{\text{CLx}}$	126- 43 35
$\boxed{\text{STO}} \boxed{1}$	119- 44 25	$\boxed{g} \boxed{\text{RTN}}$	127- 43 32
$\boxed{f} \boxed{\text{LBL}} 2$	120-42,21, 2	$\boxed{f} \boxed{\text{LBL}} 9$	128-42,21, 9
$\boxed{\text{RCL}} \boxed{(i)}$	121- 45 24	$\boxed{\text{COS}}$	129- 24
$\boxed{\text{R/S}}$	122- 31	$\boxed{\text{CHS}}$	130- 16
$\boxed{f} \boxed{\text{ISG}}$	123- 42 6	$\boxed{g} \boxed{\text{COS}^\circ}$	131- 43 24

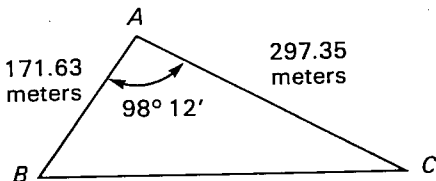
レジスタ			R _i : 間接指定
R ₀ : 面積	R ₁ : S ₁	R ₂ : A ₁	R ₃ : S ₂
R ₄ : A ₂	R ₅ : S ₃	R ₆ : A ₃	R ₇ -R ₈ : 未使用

ステップ	内容説明	入力 数値/単位	キー操作	表示 数値/単位
1	プログラムをキーイン			
2	USER モードにする			
3	角度単位を指定する			
	[DEG, RAD, GRAD].			

ステップ	操作説明	入力 数値/単位	キー操作	表示 数値/単位
4	SSS (3辺既知)			
	入力			
	辺1	s_1	<input type="button" value="ENTER"/>	
	辺2	s_2	<input type="button" value="ENTER"/>	
	辺3	s_3	<input type="button" value="A"/>	面積
	ステップ9へ			
5	ASA (2角挟辺)			
	入力			
	角3	A_3	<input type="button" value="ENTER"/>	
	辺1	s_1	<input type="button" value="ENTER"/>	
	角1	A_1	<input type="button" value="B"/>	面積
	ステップ9へ			
6	SAA (2角隣辺)			
	入力			
	辺1	s_1	<input type="button" value="ENTER"/>	
	角1	A_1	<input type="button" value="ENTER"/>	
	角2	A_2	<input type="button" value="C"/>	面積
	ステップ9へ			
7	SAS (2辺挟角)			
	入力			
	辺1	s_1	<input type="button" value="ENTER"/>	
	角1	A_1	<input type="button" value="ENTER"/>	
	辺2	s_2	<input type="button" value="D"/>	面積
	ステップ9へ			
8	SSA (2辺隣角)			
	入力			
	辺1	s_1	<input type="button" value="ENTER"/>	
	辺2	s_2	<input type="button" value="ENTER"/>	

ステップ	内容説明	入力 数値/単位	キー操作	表示 数値/単位
	角 2	A_2	[E]	面積
9	辺の長さと角度			
	3 辺と 3 角を見るとき			
			[R/S]	S_1
			[R/S]	A_1
			[R/S]	S_2
			[R/S]	A_2
			[R/S]	S_3
			[R/S]	A_3
9a	別の答がないとき		[R/S]	0
9b	別の答があるとき		[R/S]	2
			[R/S]	面積
			[R/S]	S_1
			[R/S]	A_1
			[R/S]	S_2
			[R/S]	A_2
			[R/S]	S_3
			[R/S]	A_3

例 1 ある測量士が三角形の土地を測量しています。A 点から B 点までと A 点から C 点までの距離を測定しました。AB と AC との間の角度はその前に測定してあります。これで面積と残りの辺と角を求めてください。



これは二辺挟角の問題です

$$S_1 = 171.63, \quad A_1 = 98^\circ 12', \quad S_2 = 297.35.$$

キー操作

表示

g **DEG**

f **FIX** 2

USER モードにする

171.63 **ENTER**

171.63

98.12 **g** **→H**

98.20

297.35 **D**

25,256.21

面積

R/S

171.63

S_1

R/S

98.20

A_1

R/S

297.35

S_2

R/S

27.83

A_2

R/S

363.91

S_3

R/S

53.97

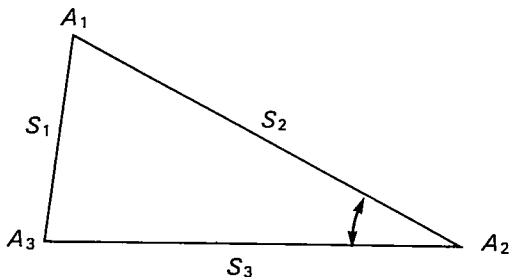
A_3

例2 2辺と隣の角から三角形を決めてください。

$$S_1 = 25.6$$

$$S_2 = 32.8$$

$$A_2 = 42.3^\circ$$



キー操作

表示

25.6 ENTER	25.60	
32.8 ENTER	32.80	
42.3 E	410.85	面積
R/S	25.60	S_1
R/S	78.12	A_1
R/S	32.80	S_2
R/S	42.30	A_2
R/S	37.22	S_3
R/S	59.58	A_3
R/S	2.00	第二の答がある
R/S	124.68	面積
R/S	25.60	S_1
R/S	17.28	A_1
R/S	32.80	S_2
R/S	42.30	A_2
R/S	11.30	S_3
R/S	120.42	A_3

t 検定

対応のあるデータの t 検定

対応のあるデータがどちらも正規分布していてその母平均が μ_1, μ_2 (未知) とします。

x_i	x_1	$x_2 \dots x_n$
y_i	y_1	$y_2 \dots y_n$

$D_i = x_i - y_i$ とすると

$$\bar{D} = \frac{1}{n} \sum_{i=1}^n D_i$$

$$s_D = \sqrt{\frac{\sum D_i^2 - \frac{1}{n} (\sum D_i)^2}{n-1}}$$

t 検定値

$$t = \frac{\bar{D}}{s_D} \sqrt{n}$$

自由度(df)は $n - 1$ です。これから $H_0: \mu_1 = \mu_2$ であるという帰無仮説の検定ができます。

二つの平均値の差の検定

正規分布している二つの母集団の母平均が μ_1, μ_2 (どちらも未知) で分散 σ^2 (未知) が等しいときにそれぞれの母集団から無作為に採取したデータを $\{x_1, x_2, \dots, x_{n_1}\}$ $\{y_1, y_2, \dots, y_{n_2}\}$ とします。次のようにして $H_0: \mu_1 - \mu_2 = d$ という帰無仮説を検定することができます。

計算式

$$\bar{x} = \frac{1}{n_1} \sum_{i=1}^{n_1} x_i$$

$$\bar{y} = \frac{1}{n_2} \sum_{i=1}^{n_2} y_i$$

$$t = \frac{\bar{x} - \bar{y} - d}{\sqrt{\frac{1}{n_1} + \frac{1}{n_2}} \sqrt{\frac{\sum x_i^2 - n_1 \bar{x}^2 + \sum y_i^2 - n_2 \bar{y}^2}{n_1 + n_2 - 2}}}$$

求めた t の自由度 (df) は $(n_1 + n_2 - 2)$ なので、帰無仮説を検定できます。

参考文献

1. **Statistics in Research**, B. Ostle, Iowa State University Press, 1963.
2. **Statistical Theory and Methodology in Science and Engineering**, K.A. Brownlee, John Wiley and Sons, 1965.

キー操作	表示	キー操作	表示
\bar{f} CLEAR [PRGM]	000-	STO 0 2	029- 44 .2
\bar{f} [LBL] [A]	001-42,21,11	RCL 0	030- 45 0
-	002- 30	STO 0 0	031- 44 .0
$\Sigma+$	003- 49	0	032- 0
R/S	004- 31	\bar{f} CLEAR Σ	033- 42 32
\bar{f} [LBL] [C]	005-42,21,13	R/S	034- 31
\bar{g} \bar{x}	006- 43 0	\bar{f} [LBL] [D]	035-42,21,14
STO 6	007- 44 6	STO 6	036- 44 6
R/S	008- 31	RCL 0 1	037- 45 .1
\bar{g} s	009- 43 48	RCL 0 2	038- 45 .2
R/S	010- 31	STO 4	039- 44 4
RCL 6	011- 45 6	$x \geq y$	040- 34
RCL 0	012- 45 0	STO 3	041- 44 3
\sqrt{x}	013- 11	\bar{g} \bar{x}	042- 43 0
x	014- 20	STO 8	043- 44 8
$x \geq y$	015- 34	$x \geq y$	044- 34
\div	016- 10	RCL 0	045- 45 0
R/S	017- 31	x	046- 20
RCL 0	018- 45 0	RCL 0 0	047- 45 .0
1	019- 1	\div	048- 10
-	020- 30	STO 7	049- 44 7
R/S	021- 31	$x \geq y$	050- 34
GTO [C]	022- 22 13	-	051- 30
\bar{f} [LBL] [B]	023-42,21,12	RCL 6	052- 45 6
$\Sigma+$	024- 49	-	053- 30
R/S	025- 31	RCL 4	054- 45 4
RCL 1	026- 45 1	RCL 3	055- 45 3
STO 0 1	027- 44 .1	RCL 7	056- 45 7
RCL 2	028- 45 2	x	057- 20

キー操作	表示	キー操作	表示
\square	058- 30	\sqrt{x}	072- 11
\square 2	059- 45 2	\square 0	073- 45 0
\square	060- 40	$1/x$	074- 15
\square 1	061- 45 1	\square 0	075- 45 .0
\square 8	062- 45 8	$1/x$	076- 15
\times	063- 20	\square	077- 40
\square	064- 30	\sqrt{x}	078- 11
\square 0	065- 45 0	\times	079- 20
\square 0	066- 45 .0	\div	080- 10
\square	067- 40	R/S	081- 31
2	068- 2	\square 9	082- 45 9
\square	069- 30	R/S	083- 31
\square 9	070- 44 9	\square 6	084- 45 6
\div	071- 10	\square D	085- 22 14

レジスタ			R ₁ : 未使用
R ₀ : n n_1, n_4	R ₁ : $\sum d$ $\sum x \sum y$	R ₂ : $\sum d^2$ $\sum x^2, \sum y^2$	R ₃ : $\sum x$
R ₄ : $\sum x^4$	R ₅ : 未使用	R ₆ : d	R ₇ : x
R ₈ : y	R ₉ : df	R ₀ : n_1	R ₁ : $\sum x$
R ₂ : $\sum x^4$	R ₃ -R ₆ :		

各レジスタの縦線の左側は対応のあるデータのあるデータの t 検定，右側は二つの平均値の差の検定のときのレジスタの内容です。

ステップ	内容説明	入力 数値/単位	キー操作	表示 数値/単位
	対応のあるデータの t 検定			
1	プログラム (001-022 ライン) をキーイン			
2	計算の準備		\square CLEAR \square REG	
3	USER モードにする			
	$i = 1, 2, \dots, n$ についてステップ 4~5 を繰返す			

ステップ	操作説明	入力 数値/単位	キー操作	表示 数値/単位
4	入力			
		x_i	ENTER	
		y_i	A	i
5	(任意操作)間違えて入力した x_k または y_k を削除するとき	x_k	ENTER	
		y_k	[-] [G] [Σ-]	$k-1$
6	検定値の計算			
	\bar{d}		C	\bar{d}
	s_D		R/S	s_D
	t		R/S	t
	df		R/S	df
7	答を見なおすときはステップ6へ 二つの平均値の差の検定			
1	プログラム(023-085 ライン)をキーイン			
2	計算の準備		f CLEAR [REG]	
3	USER モードにする			
	$i = 1, 2, \dots, n_1$ についてステップ4~5を繰返す			
4	x_i の入力	x_i	B	i
5	間違えて入力した x_k を削除するとき	x_k	[G] [Σ-]	$k-1$
6	第二の組のデータを入れる準備		R/S	
	$i = 1, 2, \dots, n_2$ についてステップ7~8を繰返す			
7	y_i の入力	y_i	B	i
8	間違えて入力した y_k を削除するとき	y_k	[G] [Σ-]	$k-1$
9	d を入力して検定値を求める			
		d	D	t
			R/S	df
10	答をもう一度見るときはステップ9へ			

例 1

x_i	14	17.5	17	17.5	15.4
y_i	17	20.7	21.6	20.9	17.2

このデータから D , s_D , t と df を求めてください。

キー操作	表示	
$\boxed{f} \boxed{FIX} 4$		
USER モードにする		
$\boxed{f} \boxed{CLEAR} \boxed{REG}$		
14 \boxed{ENTER} 17 \boxed{A}	1.0000	
17 \boxed{ENTER} 15 \boxed{A}	2.0000	間違えた
17 \boxed{ENTER}		
15 $\boxed{-} \boxed{g} \boxed{\Sigma-}$	1.0000	削除
17.5 \boxed{ENTER} 20.7 \boxed{A}	2.0000	
17 \boxed{ENTER} 21.6 \boxed{A}	3.0000	
17.5 \boxed{ENTER} 20.9 \boxed{A}	4.0000	
15.4 \boxed{ENTER} 17.2 \boxed{A}	5.0000	
\boxed{C}	-3.2000	\bar{D}
$\boxed{R/S}$	1.0000	s_D
$\boxed{R/S}$	-7.1554	t
$\boxed{R/S}$	4.0000	df

例 2

x	79	84	108	114	120	103	122	120		
y	91	103	90	113	108	87	100	80	99	54

このデータから $d = 0$ (つまり $H_0: \mu_1 = \mu_2$) として t と df を求めてください

キー操作	表示	
$\boxed{f} \boxed{CLEAR} \boxed{REG}$		
79 \boxed{B} 84 \boxed{B} 99		
$\boxed{B} 99 \boxed{g} \boxed{\Sigma-}$	2.0000	
108 \boxed{B} 114 \boxed{B}		
120 \boxed{B} 103 \boxed{B}		
122 \boxed{B} 120 \boxed{B}	8.0000	
$\boxed{R/S}$	0.0000	
91 \boxed{B} 103 \boxed{B}		

キー操作 表示

90	[B]	113	[B]		
108	[B]	87	[B]		
100	[B]	80	[B]		
99	[B]	54	[B]	10.0000	
0	[D]			1.7316	<i>t</i>
[R/S]				16.0000	<i>df</i>

χ^2 (カイ二乗) 検定

このプログラムは次の式を使って適合度 (あてはまりの良さ) の検定用の χ^2 を求めます。

$$\chi_1^2 = \sum_{i=1}^n \frac{(O_i - E_i)^2}{E_i} \quad (df = n - 1)$$

ここで O_i : 観測値 (観察値)

E_i : 期待値 (理論値)

期待値が等しいとき

$$E = E_i = \frac{\sum O_i}{n}$$

このとき

$$\chi_2^2 = \frac{n \sum O_i^2}{\sum O_i} - \sum O_i$$

注：適合度の検定である区間の期待値が小さい (5 未満) ときには、幾つかの区間をまとめて 5 以上になるようにしてください。

参考文献

Mathematical Statistics, J.E. Freund, Prentice Hall, 1962.

キー操作	表示	キー操作	表示
\boxed{f} CLEAR \boxed{PRGM}	000-	\boxed{GTO} \boxed{C}	022- 22 13
\boxed{f} \boxed{LBL} \boxed{A}	001-42,21,11	\boxed{f} \boxed{LBL} \boxed{B}	023-42,21,12
\boxed{g} \boxed{CF} 0	002-43, 5, 0	$\boxed{\Sigma+}$	024- 49
\boxed{f} \boxed{LBL} 0	003-42,21, 0	$\boxed{R/S}$	025- 31
\boxed{ENTER}	004- 36	\boxed{f} \boxed{LBL} \boxed{D}	026-42,21,14
\boxed{R} \downarrow	005- 33	\boxed{RCL} 0	027- 45 0
$\boxed{-}$	006- 30	\boxed{RCL} 2	028- 45 2
\boxed{g} $\boxed{x^2}$	007- 43 11	$\boxed{\times}$	029- 20
\boxed{g} \boxed{R} \uparrow	008- 43 33	\boxed{RCL} 1	030- 45 1
$\boxed{+}$	009- 10	$\boxed{\div}$	031- 10
\boxed{g} $\boxed{F?}$ 0	010-43, 6, 0	\boxed{RCL} 1	032- 45 1
\boxed{CHS}	011- 16	$\boxed{-}$	033- 30
\boxed{STO} $\boxed{+}$ 1	012-44,40, 1	$\boxed{R/S}$	034- 31
1	013- 1	\boxed{RCL} 1	035- 45 1
\boxed{g} $\boxed{F?}$ 0	014-43, 6, 0	\boxed{RCL} 0	036- 45 0
\boxed{CHS}	015- 16	$\boxed{\div}$	037- 10
\boxed{STO} $\boxed{+}$ 0	016-44,40, 0	$\boxed{R/S}$	038- 31
\boxed{RCL} 0	017- 45 0	\boxed{GTO} \boxed{D}	039- 22 14
$\boxed{R/S}$	018- 31	\boxed{f} \boxed{LBL} \boxed{E}	040-42,21,15
\boxed{f} \boxed{LBL} \boxed{C}	019-42,21,13	\boxed{g} \boxed{SF} 0	041-43, 4, 0
\boxed{RCL} 1	020- 45 1	\boxed{GTO} 0	042- 22 0
$\boxed{R/S}$	021- 31		

レジスタ			R ₁ : 未使用
R ₀ : カウンタ	R ₁ : $x_2^2, \Sigma O_i$	R ₂ : ΣO_i^2	R ₃ : 使用
R ₄ : 使用	R ₅ : 使用	R ₆ -R ₉ : 未使用	

ステップ	操作説明	入力 数値/単位	キー操作	表示 数値/単位
	期待値が等しくないとき			
1	プログラムをキーイン			
2	計算の準備		\boxed{f} CLEAR $\boxed{\text{REG}}$	
			\boxed{g} $\boxed{\text{CF}}$ 0	
3	USER モードにする			
	$i=1, 2, \dots, n$ についてステップ 4 ~ 5 を繰り返す			
4	入力			
	O_i	O_i	$\boxed{\text{ENTER}}$	
	E_i	E_i	$\boxed{\text{A}}$	i
5	間違えて入れた O_k また は E_k を削除するとき	O_k E_k	$\boxed{\text{ENTER}}$ $\boxed{\text{E}}$	$k-1$
6	x_1^2 の計算		$\boxed{\text{C}}$	x_1^2
	期待値が等しいとき			
1	計算の準備		\boxed{f} CLEAR $\boxed{\text{REG}}$	
			\boxed{g} $\boxed{\text{CF}}$ 0	
2	USER モードにする			
	$i=1, 2, \dots, n$ についてステップ 3 ~ 4 を繰り返す			
3	O_i 入力	O_i	$\boxed{\text{B}}$	i
4	間違えて入れた O_h を 削除するとき	O_h	\boxed{g} $\boxed{\Sigma-}$	$h-1$
5	計算			
	x_2^2		$\boxed{\text{D}}$	x_2^2
	E		$\boxed{\text{R/S}}$	E
6	答をもう一度見たいときはステップ 5 へ			

例1 次のデータが期待値にうまくあてはまっているかどうかを検定するために x^2 を求めてください。

O_i	8	50	47	56	5	14
E_i	9.6	46.75	51.85	54.4	8.25	9.15

キー操作 表示

f CLEAR REG

g CF 0

f FIX 4

USER モードにする

8 ENTER 9.6 A 1.0000

50 ENTER 46.75 A 2.0000

47 ENTER 51.85 A 3.0000

56 ENTER 54.4 A 4.0000

100 ENTER A 5.0000

100 ENTER E 4.0000

間違えた
削除

5 ENTER 8.25 A 5.0000

14 ENTER 9.15 A 6.0000

C 4.8444

 x_1^2

例2 下表はあるさいころを120回投げて、出た目の回数を調べた結果です。 χ^2 検定を使ってさいころが正常(出る目にくせがない)かどうか検討してください。自由度は5です。注 この場合はさいころが正常であることを期待しているので期待値が等しい場合のほうを使います。

目	1	2	3	4	5	6
出た回数	25	17	15	23	24	16

キー操作 表示

f CLEAR REG

g CF 0

25 B 17 B 19 B 3.0000

19 g Σ- 2.0000

間違えた
削除

15 B 23 B 24 B

16 B 6.0000

D 5.0000

R/S 20.0000

 χ_2^2
E

χ^2 分布表を見ると、自由度が5で危険率(有意水準)が5%のときは11.070です。求めた χ^2 の値は5.00で11.070よりも小さいので、このさいころはくせがあるとは言えません(くせがないということとは違います)。

複利計算 (積立, ローン, 元利合計など)

このプログラムは各種の複利計算の金額と期間について計算します。

次の項目（ただし利率だけは必ず入力）を入力したり求めたりできます。

- n 複利計算の期間。（例えば30年間の毎月返済なら $n = 12 \times 30 = 360$ です。）
- i 1計算期間の利率で、%表示で入力。（年1回の計算でなければ、年利を年間の計算回数で割った値です。例えば年利10%で毎月返済なら10/12%または0.83%を入力する。）

PV 複利計算の初回の金額（元金、貸借金額、初回投資額など）。

PMT 各回の受入・支払額

FV 最終金額（積立や定期の元利合計、貸借の最終回受入・支払額または計算期間終了時の残額など）。

各回の利息の合計額や期間途中の残額も計算できます。

このプログラムは各計算期間ごとの前払い（期首払い）と後払い（期末払い）の両方に使えるようにしてあります。後払いは普通のローンや貸付などに、前払いはリースに使っています。後払いは \boxed{A} を押した後に $\boxed{R/S}$ を押して1を表示させます。前払いは \boxed{A} を押した後に $\boxed{R/S}$ を押して0を表示させます。1度セットすれば支払法を変えなければこの操作は不要です。

新しい計算をするときには \boxed{f} CLEAR \boxed{REG} と押すのが確実で簡単な方法です。

しかし前回の計算時の入力データや答のどれかをそのまま使うときには \boxed{f} CLEAR \boxed{REG} の操作は不要です。つまり n , i , PV , PMT , FV の内のどれかをそのまま使うのならレジスタをクリアしない方がよいわけです。 \boxed{f} CLEAR \boxed{REG} を押さしないで、この項目のどれかを消すにはそこに0を入力すればよいでしょう。例えば n , i , PMT , PV の組合せの計算した後に、 n , i , PV , FV から PMT を計算するときには PMT に0をストアする必要があります。右ページの表は以上を簡単にまとめたものです。

計算できる種類

(利率がわかっているとき)

項目の組合せ	支払い・受入れ		計算の準備
	期末払い	期首払い	
n, PV, PMT この内の二つ を入力して残 りを求める	元利均等返済 ローン	リース	<input type="checkbox"/> CLEAR <input type="checkbox"/> REG を押すか FV を 0にする
n, PV, PMT, FV この内の三つ を入力して残 りを求める	最終回一括払 い併用の元利 均等返済ロー ン	期間満了時に 第3者へ転売 する予定のリ ース	不要
n, PMT, FV この内の二つ を入力して残 りを求める	減債基金	定期積立 期首払い年金	<input type="checkbox"/> CLEAR <input type="checkbox"/> REG を押すか FV を 0にする
n, PV, FV この内の二つ を入力して残 りを求める	支払時期は無関係		<input type="checkbox"/> CLEAR <input type="checkbox"/> REG を押すか PMT を0にする

計算式

$$-PV = \frac{PMT}{i} A [1 - (1+i)^{-n}] + (FV)(1+i)^{-n}$$

ただし

$$A = \begin{cases} 1 & \text{期末払い} \\ (1+i) & \text{期首払い} \end{cases}$$

注

- 利率が0のときは**Error 0**の表示になります。
- n や i が大き過ぎ (10^6) たり小さ過ぎ (10^{-6}) ると正しい答が求まりません。

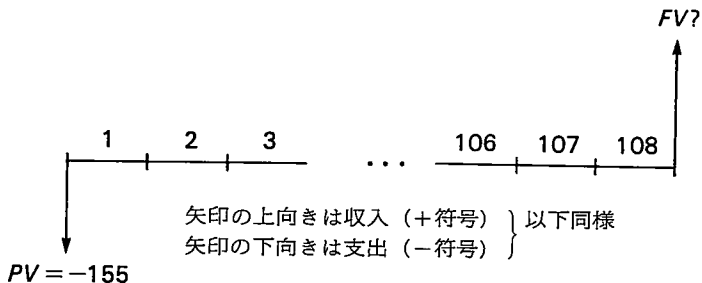
キー操作	表示	キー操作	表示
\boxed{f} CLEAR \boxed{PRGM}	000-	\boxed{STO} 1	029- 44 1
\boxed{f} LBL \boxed{A}	001-42,21,11	\boxed{g} \boxed{RTN}	030- 43 32
\boxed{f} \boxed{FIX} 2	002-42, 7, 2	\boxed{f} LBL \boxed{C}	031-42,21,13
\boxed{STO} 2	003- 44 2	\boxed{STO} 3	032- 44 3
\boxed{f} LBL 4	004-42,21, 4	$\boxed{R/S}$	033- 31
$\boxed{R/S}$	005- 31	\boxed{GSB} 1	034- 32 1
0	006- 0	\boxed{GSB} 2	035- 32 2
\boxed{g} $\boxed{F7}$ 0	007-43, 6, 0	\boxed{CHS}	036- 16
1	008- 1	\boxed{STO} 3	037- 44 3
\boxed{g} \boxed{CF} 0	009-43, 5, 0	\boxed{g} \boxed{RTN}	038- 43 32
\boxed{g} $\boxed{x=0}$	010- 43 40	\boxed{f} LBL \boxed{D}	039-42,21,14
\boxed{g} \boxed{SF} 0	011-43, 4, 0	\boxed{STO} 4	040- 44 4
\boxed{GTO} 4	012- 22 4	$\boxed{R/S}$	041- 31
\boxed{f} LBL \boxed{B}	013-42,21,12	1	042- 1
\boxed{STO} 1	014- 44 1	\boxed{STO} 4	043- 44 4
$\boxed{R/S}$	015- 31	\boxed{GSB} 1	044- 32 1
\boxed{GSB} 1	016- 32 1	$\boxed{1/x}$	045- 15
\boxed{RCL} 5	017- 45 5	\boxed{RCL} 3	046- 45 3
\boxed{g} \boxed{LSTx}	018- 43 36	\boxed{GSB} 2	047- 32 2
$\boxed{-}$	019- 30	\boxed{x}	048- 20
\boxed{RCL} 3	020- 45 3	\boxed{CHS}	049- 16
\boxed{g} \boxed{LSTx}	021- 43 36	\boxed{STO} 4	050- 44 4
$\boxed{+}$	022- 40	\boxed{g} \boxed{RTN}	051- 43 32
$\boxed{\div}$	023- 10	\boxed{f} LBL \boxed{E}	052-42,21,15
\boxed{CHS}	024- 16	\boxed{STO} 5	053- 44 5
\boxed{g} \boxed{LN}	025- 43 12	$\boxed{R/S}$	054- 31
\boxed{RCL} 6	026- 45 6	\boxed{GSB} 1	055- 32 1
\boxed{g} \boxed{LN}	027- 43 12	\boxed{RCL} 3	056- 45 3
$\boxed{\div}$	028- 10	$\boxed{+}$	057- 40

キー操作	表示	キー操作	表示
$\boxed{\text{RCL}} 7$	058- 45 7	$\boxed{y^x}$	076- 14
$\boxed{+}$	059- 10	$\boxed{\text{STO}} 7$	077- 44 7
$\boxed{\text{CHS}}$	060- 16	1	078- 1
$\boxed{\text{STO}} 5$	061- 44 5	$\boxed{x \geq y}$	079- 34
$\boxed{g} \boxed{\text{RTN}}$	062- 43 32	$\boxed{-}$	080- 30
$\boxed{f} \boxed{\text{LBL}} 1$	063-42,21, 1	$\boxed{\text{RCL}} 4$	081- 45 4
1	064- 1	$\boxed{\text{RCL}} 8$	082- 45 8
$\boxed{\text{RCL}} 2$	065- 45 2	$\boxed{+}$	083- 10
$\boxed{g} \boxed{\%}$	066- 43 14	$\boxed{\text{RCL}} 0$	084- 45 0
$\boxed{\text{STO}} 8$	067- 44 8	$\boxed{\times}$	085- 20
1	068- 1	$\boxed{\times}$	086- 20
$\boxed{\text{STO}} 0$	069- 44 0	$\boxed{g} \boxed{\text{RTN}}$	087- 43 32
$\boxed{+}$	070- 40	$\boxed{f} \boxed{\text{LBL}} 2$	088-42,21, 2
$\boxed{\text{STO}} 6$	071- 44 6	$\boxed{\text{RCL}} 5$	089- 45 5
$\boxed{g} \boxed{F?} 0$	072-43, 6, 0	$\boxed{\text{RCL}} 7$	090- 45 7
$\boxed{\text{STO}} 0$	073- 44 0	$\boxed{\times}$	091- 20
$\boxed{\text{RCL}} 1$	074- 45 1	$\boxed{+}$	092- 40
$\boxed{\text{CHS}}$	075- 16		

レジスタ			R_1 : 未使用
R_0 : 1または $1+i$	R_1 : n	R_2 : $i(\%)$	R_3 : PV
R_4 : PMT	R_5 : FV	R_6 : $1+i$	R_7 : $(1+i)^{-n}$
R_8 : $i/100$	R_9 - R_4 : 未使用		

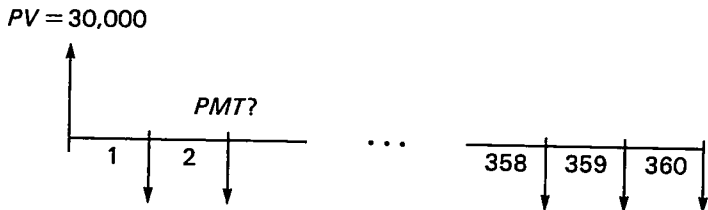
ステップ	内容説明	入力 数値/単位	キー操作	表示 数値/単位
1	プログラムをキーイン			
2	USER モードにする			
3	計算の準備.		f CLEAR REG	
4	1 期間の利率入力	$i(\%)$	A	$i(\%)$
5	期末払いにする		R/S	1.00
	または期首払いにする		R/S	0.00
	R/S を押すと交互 に切替わる			
6	わかっている値の入力			
	● 期間数	n	B	n
	● 初回金額	PV	C	PV
	● 各回支払い額	PMT	D	PMT
	● 最終値	FV	E	FV
7	未知数の計算			
	● 期間数		B R/S	n
	● 初回金額		C R/S	PV
	● 各回支払い額		D R/S	PMT
	● 最終値		E R/S	FV
8	一部を変えて計算をするときにはス テップ4へ戻り必要な数値を入れる。 場合によっては計算の前に0を入れ ることが必要 (187 ページ参照)			
9	新しい計算はステップ3へ			

例1 155ドルを年利5.75%の毎月複利で預けておくと9年後の元利合計は幾らでしょうか。



キー操作	表示	
\boxed{f} CLEAR \boxed{REG}		
USER モードにする		
5.75 \boxed{ENTER} 12 $\boxed{\div}$ \boxed{A}	0.48	月利 (%) を計算して入力
$\boxed{R/S}$ ($\boxed{R/S}$)	0.00	期首払い
9 \boxed{ENTER} 12 $\boxed{\times}$ \boxed{B}	108.00	月数
155 \boxed{CHS} \boxed{C}	-155.00	元金
\boxed{E} $\boxed{R/S}$	259.74	(元利合計)

例2 銀行から 30,000 ドルを年利 13% で 30 年間の毎月返済というこ
 とで借りました。毎月の銀行への返済額はいくらになるのでしょうか。

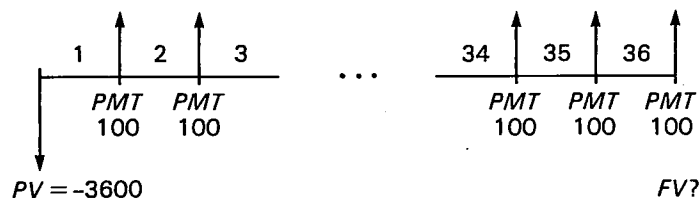


キー操作 表示

f CLEAR REG		
13 ENTER 12 ÷ A	1.08	月利 (%)
R/S (R/S)	1.00	期末払い
30 ENTER 12 × B	360.00	<i>n</i>
30000 C	30,000.00	<i>PV</i>
D R/S	-331.86	<i>PMT</i>

例3 ある人に最終回増額返済で貸すことになりました。貸出金は3,600ドル、年利10%で36月間毎月100ドルずつ返してもらうことになりました。最終回(36回目)に幾ら受取ったらよいのでしょうか。

(下の図は貸手の方から見たお金の出入の図です。借手から見れば矢印の向きが逆になります。)



キー操作 表示

f CLEAR REG		
10 ENTER 12 ÷ A	0.83	月利 (%)
R/S (R/S)	1.00	期末払い
36 B	36.00	<i>n</i>
3600 CHS C	-3,600.00	<i>PV</i>
100 D E R/S	675.27	<i>FV</i>

最終回には毎月の100ドルの他に675.27ドルがありますから合計で775.27ドル受取ることになります

例4 このプログラムはローンの支払利息の合計や途中のローン残額も計算できます。支払利息の合計は計算期間の始め n_1 と終り n_2 を指定して返済額の合計からローン残額の減少分を引きます。ローン残額の減少分は n_1 と n_2 のローン残額の差です。

例えば 50,000 ドルを年利 14% で 360 カ月の月賦返済のときの 24 回目支払後のローン残額と、13~24 カ月の間 (12 回目支払後から 24 回目支払分まで) の支払利息合計を計算してみましょう。

まず普通のローン計算をします。

キー操作	表示	
f CLEAR REG		
360 B	360.00	n
14 ENTER 12 ÷ A	1.17	i
R/S (R/S)	1.00	期末払い
50000 CHS C	-50,000.00	PV
D R/S	592.44	PMT

24ヶ月後の借入残額を求める

24 **B** **E** **R/S** 49,749.56

この金額をストアして、今度は12ヶ月後の借入残額を求める

STO 9
12 **B** **E** **R/S** 49,883.48

12ヶ月後から24ヶ月後までの借入残額の減少分を求める。

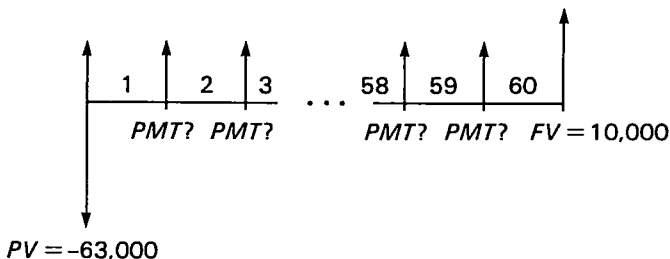
RCL 9 **-** 133.92

この期間の返済額の合計から借入残額の減少分を引くと支払利息の合計になります。

RCL 4 12 x	7,109.23	期間中の返済額の合計
x\geqy -	6,975.31	支払利息の合計

例5 あるリース会社が63,000ドルのミニコンを購入して得意先に5年契約で賃貸して年13%の利回りになるようにしたいと検討しています。リース期間満了後にそのミニコンは最低10,000ドルで売れる

見込みです。(リースなので期首払いになります。) 年利13%の利回りにするためにはリース料は月額幾らにしたらいでしょうか。



キー操作

表示

f CLEAR REG		
13 ENTER 12 ÷ A	1.08	<i>i</i>
R/S	0.00	期首払い
5 ENTER 12 x B	60.00	<i>n</i>
63000 CHS C	-63,000.00	<i>FV</i>
10000 E D R/S	1,300.16	<i>PMT</i>

仕入金額が70,000ドルではどうでしょうか。

70000 CHS C	-70,000.00	
D R/S	1,457.73	<i>PMT</i>

潜水艦撃沈ゲーム

10×10のます目のゲーム盤に隠れている潜水艦を探知機(ソナー)で探だし、爆雷で撃沈するゲームです。

まず乱数の種(0から1の間の数)をキーインして**E**を押すと、ゲーム盤の100個のます目(縦位置が0から9まで、横位置も0から9までの交点)のどこかの中心に潜水艦を隠します。

次に潜水艦の隠れていそうなところを予想します。縦と横位置を指定して（移動して）探知機を作動させるために [B] を押します。移動した位置とその周りの8個のます目に潜水艦が隠れていれば、探知機に反応があって 1 を表示します。探知機に反応がなければ 0 の表示になります。

探知機を何回か作動させながら潜水艦の真上に移動して、爆雷を投下し（[A] を押し）ます。1 が点滅したら爆雷がうまく命中したことになり、0 だったら命中しなかったことになります。命中しなかったら潜水艦はその位置から上下左右のどれか隣のます目（縦か横位置だけが同じ）に逃げてしまいます。

ゲームをもっと難しくするには乱数の種を入力した後に [C] を押します。こうすると探知機を作動させるごとに、爆雷が命中しなかったときのように隣のます目に逃げてしまいます。

爆雷の有効半径は投下点から0.9ます分です。爆雷の投下位置は各ます目の中央でなくてもかまいません。例えば (2.5, 6.5) の点で爆雷を投下すると (2,6), (2,7), (3,6) と (3,7) のどれかのます目に隠れた潜水艦を撃沈できます。

探知機の操作回数は10回以内、爆雷は1発だけで潜水艦を撃沈するように挑戦してみてください。（回数の制限はありませんが。）計算途中でなければいつでも [D] を押せばゲームの進行状態を次のような形式で見られます。

XX.YY XX は爆雷の投下数, YY は探知機の操作回数

キー操作	表示	キー操作	表示
[f] CLEAR [PRGM]	000-	[g] [RTN]	005- 43 32
[f] [LBL] [C]	001-42,21,13	[f] [LBL] [E]	006-42,21,15
1	002- 1	[f] CLEAR [REG]	007- 42 34
[STO] 0	003- 44 0	[g] [CF] 0	008-43, 5, 0
[g] [SF] 0	004-43, 4, 0	[STO] [RAN#]	009- 44 36

キー操作	表示	キー操作	表示
$\boxed{\text{GSB}} 9$	010- 32 9	$\boxed{\text{f}} \boxed{\text{PSE}}$	039- 42 31
$\boxed{\text{STO}} 1$	011- 44 1	$\boxed{\text{g}} \boxed{\text{RTN}}$	040- 43 32
$\boxed{\text{GSB}} 9$	012- 32 9	$\boxed{\text{f}} \boxed{\text{LBL}} \boxed{\text{B}}$	041-42,21,12
$\boxed{\text{STO}} 2$	013- 44 2	1	042- 1
$\boxed{\text{f}} \boxed{\text{FIX}} 0$	014-42, 7, 0	$\boxed{\text{STO}} \boxed{+} 8$	043-44,40, 8
$\boxed{\text{g}} \boxed{\text{CLx}}$	015- 43 35	$\boxed{\text{R}} \boxed{\downarrow}$	044- 33
$\boxed{\text{g}} \boxed{\text{RTN}}$	016- 43 32	$\boxed{\text{f}} \boxed{\text{FIX}} 0$	045-42, 7, 0
$\boxed{\text{f}} \boxed{\text{LBL}} \boxed{\text{A}}$	017-42,21,11	$\boxed{\text{g}} \boxed{\text{CF}} 1$	046-43, 5, 1
1	018- 1	$\boxed{\text{GSB}} 6$	047- 32 6
$\boxed{\text{STO}} \boxed{+} 7$	019-44,40, 7	$\boxed{\text{RCL}} 0$	048- 45 0
$\boxed{\text{R}} \boxed{\downarrow}$	020- 33	$\boxed{\text{STO}} 5$	049- 44 5
$\boxed{\text{g}} \boxed{\text{SF}} 1$	021-43, 4, 1	$\boxed{\text{g}} \boxed{\text{F7}} 0$	050-43, 6, 0
$\boxed{\text{GSB}} 6$	022- 32 6	$\boxed{\text{GSB}} 5$	051- 32 5
$\boxed{\text{g}} \boxed{x \neq 0}$	023- 43 30	$\boxed{\text{RCL}} 3$	052- 45 3
$\boxed{\text{GTO}} 0$	024- 22 0	$\boxed{\text{g}} \boxed{\text{RTN}}$	053- 43 32
1	025- 1	$\boxed{\text{f}} \boxed{\text{LBL}} 5$	054-42,21, 5
$\boxed{\text{STO}} 5$	026- 44 5	$\boxed{\text{GSB}} 9$	055- 32 9
$\boxed{\text{GSB}} 5$	027- 32 5	4	056- 4
$\boxed{\text{g}} \boxed{\text{RTN}}$	028- 43 32	$\boxed{x \cong y}$	057- 34
$\boxed{\text{f}} \boxed{\text{LBL}} 0$	029-42,21, 0	$\boxed{\text{f}} \boxed{x > y}$	058- 42 20
9	030- 9	$\boxed{\text{GTO}} 0$	059- 22 0
$\boxed{1/x}$	031- 15	$\boxed{\text{RCL}} 5$	060- 45 5
$\boxed{\text{f}} \boxed{\text{FIX}} 3$	032-42, 7, 3	$\boxed{\text{CHS}}$	061- 16
$\boxed{\text{f}} \boxed{\text{PSE}}$	033- 42 31	$\boxed{\text{GTO}} 1$	062- 22 1
$\boxed{\text{f}} \boxed{\text{FIX}} 5$	034-42, 7, 5	$\boxed{\text{f}} \boxed{\text{LBL}} 0$	063-42,21, 0
$\boxed{\text{f}} \boxed{\text{PSE}}$	035- 42 31	$\boxed{\text{RCL}} 5$	064- 45 5
$\boxed{\text{f}} \boxed{\text{FIX}} 7$	036-42, 7, 7	$\boxed{\text{f}} \boxed{\text{LBL}} 1$	065-42,21, 1
$\boxed{\text{f}} \boxed{\text{PSE}}$	037- 42 31	$\boxed{\text{STO}} 6$	066- 44 6
$\boxed{\text{f}} \boxed{\text{FIX}} 9$	038-42, 7, 9	$\boxed{\text{GSB}} 9$	067- 32 9

キー操作	表示	キー操作	表示
5	068-	5 $\boxed{f} \boxed{FIX} 0$	097-42, 7, 0
$\boxed{f} \boxed{x > y}$	069- 42 20	$\boxed{g} \boxed{RTN}$	098- 43 32
$\boxed{GTO} 0$	070- 22 0	$\boxed{f} \boxed{LBL} 6$	099-42,21, 6
$\boxed{RCL} 1$	071- 45 1	$\boxed{RCL} 2$	100- 45 2
$\boxed{GSB} 1$	072- 32 1	$\boxed{-}$	101- 30
$\boxed{STO} 1$	073- 44 1	$\boxed{x \geq y}$	102- 34
$\boxed{GTO} 2$	074- 22 2	$\boxed{RCL} 1$	103- 45 1
$\boxed{f} \boxed{LBL} 0$	075-42,21, 0	$\boxed{-}$	104- 30
$\boxed{RCL} 2$	076- 45 2	$\boxed{g} \boxed{\rightarrow P}$	105- 43 26
$\boxed{GSB} 1$	077- 32 1	$\boxed{STO} 4$	106- 44 4
$\boxed{STO} 2$	078- 44 2	$\boxed{g} \boxed{F?} 1$	107-43, 6, 1
$\boxed{GTO} 2$	079- 22 2	$\boxed{GTO} 0$	108- 22 0
$\boxed{f} \boxed{LBL} 1$	080-42,21, 1	1	109- 1
$\boxed{RCL} 6$	081- 45 6	$\boxed{-}$	110- 30
$\boxed{+}$	082- 40	$\boxed{f} \boxed{LBL} 0$	111-42,21, 0
$\boxed{g} \boxed{x < 0}$	083- 43 10	$\boxed{\cdot}$	112- 48
$\boxed{GTO} 0$	084- 22 0	9	113- 9
9	085- 9	$\boxed{-}$	114- 30
$\boxed{x \geq y}$	086- 34	$\boxed{g} \boxed{x < 0}$	115- 43 10
$\boxed{f} \boxed{x \leq y}$	087- 42 10	$\boxed{GTO} 0$	116- 22 0
$\boxed{g} \boxed{RTN}$	088- 43 32	0	117- 0
$\boxed{f} \boxed{LBL} 0$	089-42,21, 0	$\boxed{GTO} 1$	118- 22 1
$\boxed{RCL} 6$	090- 45 6	$\boxed{f} \boxed{LBL} 0$	119-42,21, 0
2	091- 2	1	120- 1
$\boxed{\times}$	092- 20	$\boxed{f} \boxed{LBL} 1$	121-42,21, 1
$\boxed{-}$	093- 30	$\boxed{STO} 3$	122- 44 3
$\boxed{g} \boxed{RTN}$	094- 43 32	$\boxed{g} \boxed{RTN}$	123- 43 32
$\boxed{f} \boxed{LBL} 2$	095-42,21, 2	$\boxed{f} \boxed{LBL} \boxed{D}$	124-42,21,14
$\boxed{RCL} 3$	096- 45 3	$\boxed{f} \boxed{FIX} 2$	125-42, 7, 2

キー操作	表示	キー操作	表示
RCL 7	126- 45 7	f LBL 9	133-42,21, 9
RCL 8	127- 45 8	f RAN#	134- 42 36
EEX	128- 26	1	135- 1
2	129- 2	0	136- 0
÷	130- 10	x	137- 20
+	131- 40	g INT	138- 43 44
g RTN	132- 43 32	g RTN	139- 43 32

レジスタ			R _i : 未使用
R ₀ : 0, 1	R ₁ : P ₁	R ₂ : P ₂	R ₃ : 反応
R ₄ : d	R ₅ : 使用	R ₆ : 使用	R ₇ : 使用
R ₈ : 使用			

ステップ	内容説明	入力 数値/単位	キー操作	表示 数値/単位
1	プログラムをキーイン			
2	USER モードにする			
3	乱数の種(0 ≤ n ≤ 1の数)を入力する	n	E	0
4	普通のゲームはステップ6へ			
5	難しいほうのゲーム(每回潜水艦が動く)		C	1
6	探知装置作動	縦	ENTER	
	"0"は反応なし	横	B	0または1
	"1"は反応があった			
	または			
	爆電投下	縦	ENTER	
	"0"ははずれた	横	A	0または
	"1"が点滅したら命中			点滅

ステップ	内容説明	入力 数値/単位	キー操作	表示 数値/単位
7	潜水艦に命中する			
	までステップ6を繰返す			
8	探知機と爆雷の使用数を見るには		□	XX.YY
	XX = 爆雷投下数			
	YY = 探知機使用回数			
9	ゲームを繰返すときはステップ3へ			

	0	1	2	3	4	5	6	7	8	9		0	1	2	3	4	5	6	7	8	9		
9										9													9
8										8													8
7										7													7
6										6													6
5										5													5
4										4													4
3										3													3
2										2													2
1										1													1
0										0													0
	0	1	2	3	4	5	6	7	8	9		0	1	2	3	4	5	6	7	8	9		
9										9													9
8										8													8
7										7													7
6										6													6
5										5													5
4										4													4
3										3													3
2										2													2
1										1													1
0										0													0
	0	1	2	3	4	5	6	7	8	9		0	1	2	3	4	5	6	7	8	9		
9										9													9
8										8													8
7										7													7
6										6													6
5										5													5
4										4													4
3										3													3
2										2													2
1										1													1
0										0													0
	0	1	2	3	4	5	6	7	8	9		0	1	2	3	4	5	6	7	8	9		

潜水艦撃沈ゲームのゲーム盤です。これをコピーしてお使いください。

例 1

キー操作 表示

USER モードにする

.58 [E] 0.

第 1 回目

3 [ENTER] 8 [B] 1. 反応があった

下図の×印のどこかに隠れている。

	0	1	2	3	4	5	6	7	8	9	
9											9
8											8
7											7
6											6
5											5
4								X	X	X	4
3								X	X	X	3
2								X	X	X	2
1											1
0											0
:	0	1	2	3	4	5	6	7	8	9	

1 回目の結果

第 2 回目

4 [ENTER] 7 [B] 0. 反応がなかった

下図の⊗印にはいない

	0	1	2	3	4	5	6	7	8	9	
9											9
8											8
7											7
6											6
5											5
4								⊗	⊗	X	4
3								⊗	⊗	X	3
2								X	X	X	2
1											1
0											0
	0	1	2	3	4	5	6	7	8	9	

2 回目の結果

第3回目

2 [ENTER] 9 [B]

0.

反応なし

潜水艦は下図の二つの×印のどちらかにいる。

	0	1	2	3	4	5	6	7	8	9	
9											9
8											8
7											7
6											6
5											5
4								X	X	X	4
3								X	X	X	3
2								X	X	X	2
1											1
0											0
	0	1	2	3	4	5	6	7	8	9	

第4回目

4 [ENTER] 9 [B]

1.

反応あり

潜水艦が見つかったので(2,7)を探す必要がなくなった。

	0	1	2	3	4	5	6	7	8	9	
9											9
8											8
7											7
6											6
5											5
4								X	X	X	4
3								X	X	X	3
2								X	X	X	2
1											1
0											0
	0	1	2	3	4	5	6	7	8	9	

4回目の結果

第5回目

4 [ENTER] 9 [A]

0.111

0.11111

0.1111111
 0.11111111
 0.11111111 命中した。

例 2

キー操作	表示
.6 [E]	0
[C]	1.

探知機を操作するたびに爆雷が命中しなかったときのように潜水艦が逃げます。

第1回目

7 [ENTER] 4 [B] 1. 反応があった

潜水艦は下図の左側の×印のどこかに隠れています。しかし潜水艦は1ます分逃げたので右側の×印のどこかにいます。

	0	1	2	3	4	5	6	7	8	9		0	1	2	3	4	5	6	7	8	9	
9										9				X	X	X						9
8			X	X	X					8			X	X	X	X	X					8
7			X	X	X					7			X	X	X	X	X					7
6			X	X	X					6			X	X	X	X	X					6
5										5				X	X	X						5
4										4												4
3										3												3
2										2												2
1										1												1
0										0												0
	0	1	2	3	4	5	6	7	8	9		0	1	2	3	4	5	6	7	8	9	

1回目の結果

第2回目

6 [ENTER] 4 [B] 0. 反応なし

左側の(×)印にはいませんでしたが、また1ます分逃げたので右側の×印のどこかを探す必要がありますか

	0	1	2	3	4	5	6	7	8	9		0	1	2	3	4	5	6	7	8	9	
9			X	X	X					9			X	X	X	X	X					9
8			X	X	X	X	X			8		X	X	X	X	X	X	X	X			8
7			X	X	X	X	X			7		X	X	X	X	X	X	X	X			7
6			X	X	X	X	X			6		X	X	X		X	X	X				6
5				X	X	X				5			X				X					5
4										4												4
3										3												3
2										2												2
1										1												1
0										0												0
	0	1	2	3	4	5	6	7	8	9		0	1	2	3	4	5	6	7	8	9	

2回目の結果

第3回目

7 [ENTER] 3 [B]

1.

反応があった

かなり場所をしぼれたけれども、また逃げたので右側の×を調べなくてはならない。

	0	1	2	3	4	5	6	7	8	9		0	1	2	3	4	5	6	7	8	9	
9			X	X	X	X	X			9			X	X	X							9
8		X	X	X	X	X	X	X		8		X	X	X	X	X						8
7		X	X	X	X	X	X	X		7		X	X	X	X	X						7
6		X	X	X		X	X	X		6		X	X	X	X							6
5			X				X			5			X	X								5
4										4												4
3										3												3
2										2												2
1										1												1
0										0												0
	0	1	2	3	4	5	6	7	8	9		0	1	2	3	4	5	6	7	8	9	

3回目の結果

第 4 回目 ここで爆雷を投下してみましょう

8 **ENTER** 3 **A**

0.111

0.11111

0.1111111

0.111111111

0.111111111 命中した

やったあ。

潜水艦はここにいました。

	0	1	2	3	4	5	6	7	8	9	
9											9
8			X								8
7											7
6											6
5											5
4											4
3											3
2											2
1											1
0											0
	0	1	2	3	4	5	6	7	8	9	

第11章 プログラム作成の技術

プログラムの構造

構造という少し大げさですが、HP-11Cを便利にうまく使うためにプログラムのまとめかたなどを説明したいと思います。といってもプログラムを幾つもまとめた人なら知っていることばかりです。どちらかと言うと正しいプログラムをわかりやすく、早くまとめるにはどういう点に注目するかとかどう考えるかとかについて説明します。これから説明する例がお役にたつことを願っています。

問題点の把握

プログラムをまとめるときに一番初めにするのはどうやって何を求めたいのかを書きだすことです。これは非常に大切なことですが省略してしまう人がかなりいます。問題点（何を求めたいのか）がはっきりしていないとプログラム作成に慣れている人でも正しいプログラムを作れないことがあります。この問題点をはっきりさせて、どんなふうに関係式を書きだすことは非常に大切なことです。こうしなければプログラム作成を始められません。

例 a, b, c を係数としたときに $ax^2+bx+c=0$ の根を求めたいと考えたとしましょう。問題点を書きだしてみますと、“ a, b と c を入力して、 $ax^2+bx+c=0$ の二つの根を求めたい”になります。

これを読むと、二つのことがはっきりします。1)係数を入力する、2)答(根)を表示させるということです。これがはっきりすれば第二段階の計算手順(アルゴリズム)の検討に進むことができます。

計算手順(アルゴリズム)

計算手順はプログラムでなく、答を求めるための考えかたのことです。計算手順といっても専門的な用語でなく、計算の順序を一つずつ順にまとめるだけです。そして適当に行間をあけておけばプログラム命令

を書きこんだり、メモを書きこむこともできます。先程の例題をまとめてみましょう。この場合は二次式の根という次の計算式を使う必要があります。

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

この式を使うと計算手順は次のようになります。

$b^2 - 4ac$ を計算します。

この値が正数なら、 $\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$ を計算します。

この値が負数なら、 $\frac{-b}{2a}$ と $\frac{\pm \sqrt{|b^2 - 4ac|}}{2a}$ の両方を計算します。

$b^2 - 4ac$ が正数なら答は二つの実根になります。また負数なら答は二つの複素数になります。これがそのままプログラムになるわけではありませんが、答を求めるときに必要な手順の全部です。

この基本の計算手順をさらに細かくすると実際のプログラム命令に近くなってきます。このように書替えるとどのようにプログラムを組めばよいのがはっきりしてきます。上の例は次のようになります。

R_A , R_B と R_C のレジスタを使うと仮定すると

1. b を取出して正負符号を替え、 a を2倍して割る。
2. この結果を R_A にストアする。
3. b を二乗して、 a と c を掛けたものの4倍を引く、その結果を R_B にストアする。
4. R_B の絶対値の平方根を求め、 a を2倍して割る。
5. この結果を R_C にストアする。

6. R_B の値が正数か0であれば、 R_A と R_C をたして表示する（実根）。
7. R_B が0よりも小さければ（負数ならば）、 R_A と R_C を別々に表示する（虚根）。
8. R_C の値の正負符号を変えて、6または7ステップ目を繰り返す。

このように書替えると、中間結果をストアするためにレジスタを幾つ使うかもはっきりします。同様にどのような操作をするかもはっきりします。

計算手順を細かく書替えると、計算順序がはっきりしてプログラムに近くなり、プログラムへの変更が簡単になります。次の計算手順はかなりうまくまとまった例です。

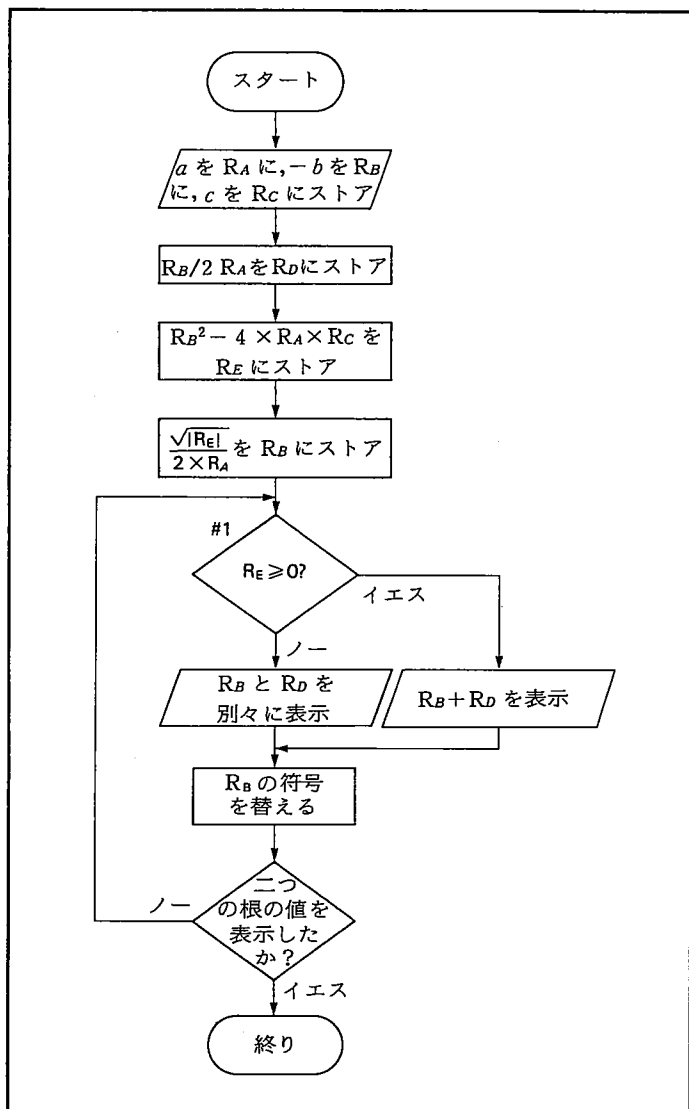
ここでは $R_A \sim R_E$ の5個のレジスタを使うものと考えました。

1. a を R_A に、 b の符号を替えて R_B に、 c を R_C にストアする。
2. R_B を R_A の2倍で割ってその答を R_D にストアする。
3. R_A と R_C を掛けて4倍し、 R_B の二乗から引いてその答を R_E にストアする。
4. R_E の絶対値の平方根を求め、 R_A の2倍で割ってその答を R_B にストアする。
5. R_E が正または0だったら R_B に R_D をたしたものを表示する。
6. R_E が負だったら R_B と R_D を別々に表示する。
7. R_B の符号を替えてステップ5または6をもう一度繰り返す。

この計算手順は R_B に最初のデータと、中間計算結果を入れるために二度使って、記憶レジスタの使用数を減らしています。

流れ図（フローチャート）

あなたがまとめた計算手順を一目でわかるようにしたものが流れ図です。計算手順とその流れを図にまとめたもので、どのように考え・どのように続くかがよくわかります。上の例を流れ図にすると右ページのようになります。



始めは難しそうに見えますが、よく見ると簡単な計算を集めたものです。流れ図を解説するには、左上方のスタートから矢印に沿って順々に見ます。どの枠も一方向の矢印でつながっています。つまり流れの方向は一方向で、普通は上から下に向っています。

途中に#1と書いてあるひし形の枠があって、ここは“条件ジャンプ”といって出口を選ぶようになっています。これからの矢印は枠の中の質問の答によって方向が変わります。この例では“REは0と等しいか、または大きいか？”です。答がイエスなら矢印は右へ進み、ノーなら矢印が下方へ進みます。

このように流れ図はジャンプやループなどのときに流れの方向をはっきりさせて間違いを防ぐのに大変役に立ちます。

サブルーチン

第10章の始めにマトリクス計算のプログラムがあって、そのプログラムがかなり長いのでびっくりしたと思います。実際にプログラム全体を同時にHP-11Cのメモリー中に入れておくことはできません。このプログラムは別ですが、このように長いプログラムをかなり短くすることはできないものでしょうか。例えば同じような計算の繰返しを何回もやっているようなときです。このマトリクス計算でも逆行列のときに同じような計算を何回もやっています。このようなときにはサブルーチンという方法を使ってプログラムを短くすることができます。

計算機でサブルーチンといえばラベル (**LBL** *n*) で始まってリターン (**RTN**) か、プログラムの最後のところで終わっています。この連続した部分をサブルーチンといって入口と出口があります。(これ以外にライン番号を指定してサブルーチンに入る方法もあります。137ページのライン番号を間接指定したジャンプとサブルーチンも見てください。)

サブルーチンを利用するには **GSB** *n* (*n* はサブルーチンの始めのラベル番号) を使います。マトリクス計算の本体は000から078ラインで、途中に **GSB** 8が8回も出てきます。**GSB** 8があるごとに **LBL**

8へジャンプし、そこから **RTN** までを順々に実行し、**RTN** で **GSB** 8の次のところに戻っています。

サブルーチンの利点はプログラムが短くなることです。それには8個所の **GSB** 8の部分にサブルーチンの **LBL** 8を入替えてみるとどんなにプログラムが長くなってしまいか比較できます。(2倍近い150ラインになってしまいます。) このようにサブルーチンは非常に便利なものです。プログラムを読解するときにはプログラムを幾つかの部分に区切ります。こうすれば全体を一度に読解するよりもずっと楽になります。各部分がそれぞれわかればプログラム全体の働きもわかるようになります。細かく区切っておくとエラー訂正も簡単になります。エラー原因の発見も、その対策もすぐに行えるようになると思います。

ISG と組合わせた **RCL** (i)

あなたが考えた計算手順をよくみると **STO** 1... **STO** 2... **STO** 3のように数値のストアとリコールを何回も繰返していることがあるかも知れません。このようなときにはHP-11Cの **STO** (i) や **RCL** (i) をサブルーチンなどに使うと簡単になります。この手法をマトリクス計算の中で有効に使っています。

LBL **A** (079から101ライン) では 3×3 の行列式の計算のために R_5 から R_3 の数値を使っています。この計算を数学的に表現すると次の通りになります。

$$\begin{vmatrix} R_5 & R_6 & R_7 \\ R_8 & R_9 & R_0 \\ R_1 & R_2 & R_3 \end{vmatrix} = \begin{aligned} & R_5(R_9 \times R_3 - R_0 \times R_2) \\ & - R_6(R_8 \times R_3 - R_0 \times R_1) \\ & + R_7(R_8 \times R_2 - R_9 \times R_1) \end{aligned}$$

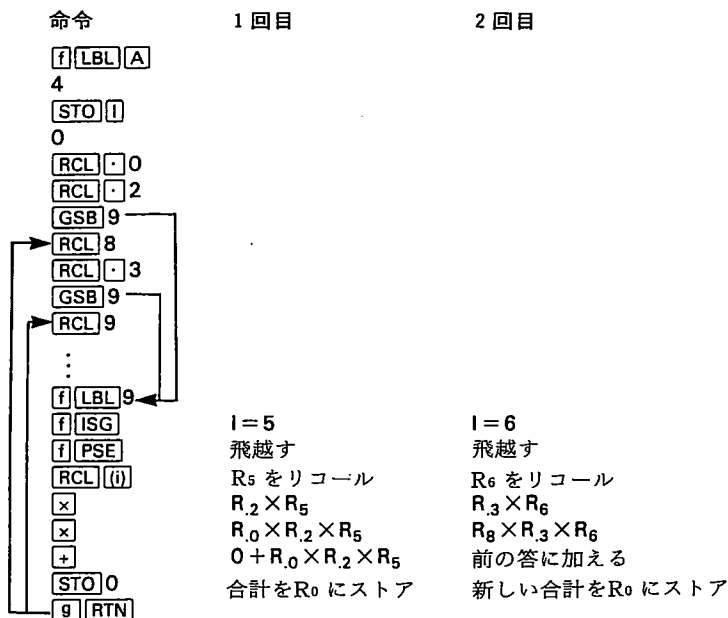
これを並べ替えると次のようになります。

$$\begin{aligned} & -(R_0 \times R_2 \times \underline{R_5} + R_8 \times R_3 \times \underline{R_6} + R_1 \times R_9 \times \underline{R_7}) \\ & + R_7 \times R_2 \times \underline{R_8} + R_5 \times R_3 \times \underline{R_9} + R_6 \times R_1 \times \underline{R_0} \end{aligned}$$

これを見ると $R \times R \times R +$ という形と、 R_5 から R_0 までを順々にリコールしているのがわかると思います。この二つを一つにまとめたのが

このサブルーチンでIレジスタ中の数値を増加させるために **ISG** を使っています。(128ページのIレジスタの増減を見てください。) このサブルーチンを使うごとに R_i が1ずつ増加するので、 R_i を使った間接リコール (**RCL** (i)) するレジスタも一つずつ変わります。このとき R_i の小数部分はいつも0であり、整数部分はいつも0より大きいので **ISG** の次のラインを飛越します。この飛越し用に何の役にも立たない **PSE** をわざと入れてあります。

このサブルーチンの使い方は下図の通りです。



これでプログラムをどう短くしたかがおわかりになると思います。1回につき4ラインで6回繰返しているので合計24ラインにもなります。これを10ラインのサブルーチン(4 **STO** **I** と **STO** **0** までを勘定に入れました)にまとめてあります。

データの入力

プログラムを作っているときにまずどんな数値（データ）が必要かと考え、次にそれをどのようにストアしようか考えたと思います。それにはプログラムのあきスペースや、何個のデータをストアするかによっていろいろな方法があります。

三角形の計算では SSS, ASA, SAA, SAS, と SSA の 5 通りについて 3 個のデータを入れました。5 種とも同じ方法ですが別々のレジスタにストアしました。HP-11C には 5 個のユーザ指定キーがあるので、5 種それぞれのキーを使い分けて 3 個のデータをそれに応じたレジスタにストアしたわけです。

ユーザ指定キーを使う別の方法もあります。その一つは 3 個のデータを手操作でそれに応じたレジスタにストアして（例えば A 1 は **STO** 2, S 2 は **STO** 3 のように）から、目的に応じたユーザ定義キーを押す方法です。この方法は長いプログラムのためにメモリーに余裕がなくて、データ数が少ないときにはよい方法ですが、一般的にはレジスタ番号を確認するなど手数と時間がかかるのであまり良い方法ではありません。

もう一つがよく使う方法は“止まってからデータをキーインする”という方法です。これもユーザ指定キーと組み合わせることがあります。ユーザ指定キーを押して計算機が止まったら一番始めのデータをキーインします。それから **R/S** を押してプログラムを続行させます。プログラムでそのデータに応じたレジスタにストアしてから次のデータ入力待ちのために止めるようにするわけです。これを全部のデータを入力し終るまで繰返すようにします。この“止まってからデータをキーインする”方法とループを組み合わせると便利なのでこれを応用するのもよいでしょう（三元連立一次方程式の根の **LBL** **A** の部分を見てください）。

三角形の計算のプログラムで利用した方法はスタックに全部のデータ（この場合では 3 個だけです）を入力しておいてから計算目的に応じたユーザ指定キーを押すようにしました。ユーザ指定キーを押すとどれも **STO** RA, **R↓**, **STO** RB, **R↓**, **STO** Rc というようにそれぞれに応じたレジスタに 3 個のデータをストアさせています。この方法はプログラムが短くてすみ、操作が速くて簡単な良い方法です。

ループ

ニュートン法のプログラムは方程式の根を求めるときに使う一般的な方法です。普通の代数的方法ではどうしても根を求められないけれども何とか正しい根を求めたいときに使います。

ニュートン法ではある初期値 x のときの $f(x)$ を求めてはその方程式の根に近づいていきます。(このときに代数などを利用して初期値の範囲を決めるともっと速くなります。) ある初期値で一度にどれだけ正解に近づくかは始めに使った初期値次第です。一つ計算をしてはその値を使って次の答を求めています。この方法を何回も繰返し、場合によっては回数を制限して、できるだけ正しい答を求めようというわけです。

プログラム中の一部に求めたい答が見つかるまで何回も同じ計算を繰返している部分があります。この部分をループと呼びます。(このプログラムのループの主な部分は 026 から 052 ラインです。) ループの繰返し回数を制限しておかないと答が求まるまでに長時間かかることがあるので、このプログラムでループを途中で打切るために繰返し回数を指定するようにしました。



ループを途中で打切る一つの方法は、ループの途中にカウンタ (レジスタを1個) を入れてループを通るごとに1ずつレジスタの値を増やそうにします。そしてこのレジスタの値と最大繰返し回数とを比較します。両方の値が等しくなったときにループを抜出するようにすればよいわけです。(このプログラムでは最大数をストアしておき、ループを通るごとに1ずつ減らして0になったらループを抜出するようにしました。051 ラインがそうです。) ただしこの方法はいつも正しい答が求まるとは限りません。初期値によっては何回も繰返しても正しい答が見つからないことがあります。100 以上繰返して求まった答が正しい答とかけ離れていることもあるわけです。

もっと良い方法は2回分の前の答と後の答の差を求めて、差の大きさによって判断する方法です。この差つまり Δx をプログラムの始めに指定しておけばよいのです。

ループにこの方法を使うには

1. Δx と初期値をストアする。
2. 初期値を使って 1 回目の近似値を求めてストアする。
3. この近似値を使って次の近似値を求めてストアする。
4. この近似値と前回の近似値の差を求める。
5. Δx を呼出して近似値の差の絶対値と比較する。
6. 差の方が Δx よりも大きかったらループを繰返す。
7. 差の方が Δx よりも小さいか等しくなったらループを抜出す。

この方法は必ずループを抜出すとは限りません。初期値によっては正しい答とかけ離れていたり、正しい答が見つからないこともあるからです。そこでこのプログラムのように回数と Δx の指定を組合せるのが一番良いでしょう。

このようにしても計算式によっては正しくない（見せかけの）答が求まることがあります。計算式の勾配が急で近似値の差が Δx 以内になってしまうと正しくない答がでる原因になります。そこで上の二つの方法以外に求まった答が正しいかどうかチェックできるようにしています。求まった答を計算式に代入して 0 との差を目で直接確認できます。

フ ラ グ

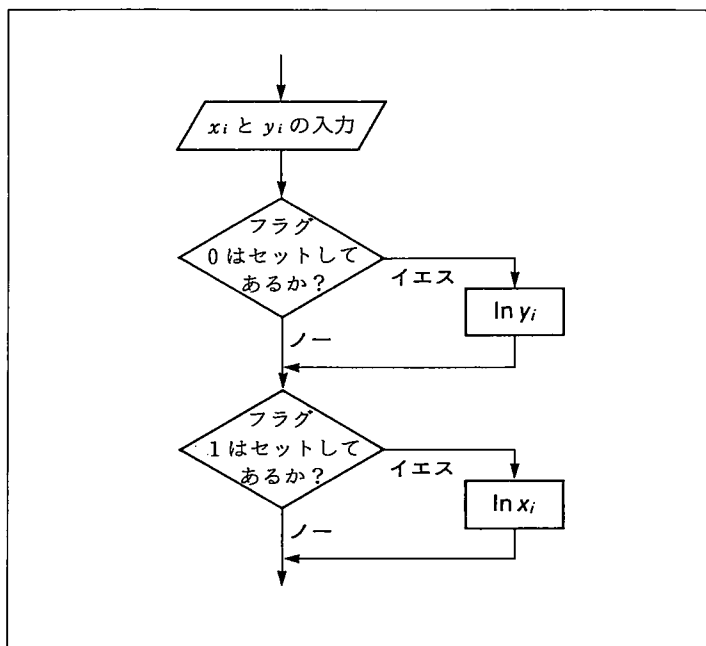
プログラムをまとめるときに似たような方法でも厳密には少し違うことがあります。この違いを分けるのにフラグの利用があります。

フラグはメモのようなものです。プログラム計算の入口の一つでフラグをセットさせるようにします。計算の途中でフラグの判断をすればどこを通過してきたか調べられます。例えばフラグ 0 をセットしてあるかというのは、フラグ 0 をセットする部分を通ってきたかということと同じことです。このようにしてフラグも条件判断に使えます。

曲線のあてはめのプログラムでは選んだ曲線の種類に応じて 3 種の似た計算を使いわけています。このフラグの使い方を見ていただくために 162 ページの表に追加すると次のようになります。

曲線の種類	A	X_i	Y_i	コード	フラグ0	フラグ1
指数	$\ln a$	x_i	$\ln y_i$	1	セット	クリア
対数	a	$\ln x_i$	y_i	2	クリア	セット
累乗	$\ln a$	$\ln x_i$	$\ln y_i$	3	セット	セット

プログラムの始めでフラグを二つともセットしておき、曲線の種類に応じて上の表のようにフラグを選んでクリアします。この二つのフラグの状態によって入力したデータ (x_i と y_i) を処理しています。



Aについても同様です。

このようにフラグを活用すると非常に便利です。他の方法だと三つの別々なプログラムが必要なところもうまく一つにまとまりました。フラグはこのように有効な手段なのでどんどん活用してください。

乱数

HP-11Cは乱数発生用のキー（**RAN#**）があります。計算機やコンピュータでは本当の乱数が作れないので、擬似一様乱数を発生させてこれを使います。機械の性質のため入れる数値によって出てくる数値が予想できます。そこでHP-11Cの乱数発生は“種（seed）”を使う人がストアして、出てくる答（乱数）が（使う人に）予想できないようにしました。答を n とすると $0 \leq n < 1$ になります。

乱数の用途は各種ありますが、一番よく使うのはゲームのプログラムでしょう。それは相手に次の数が幾つであるかわかってはつまらないからです。潜水艦撃沈ゲームは乱数を使った例です。ゲームを始めるときに種を入れるとそれから乱数を使って潜水艦を配置します。

このプログラムでは整数の乱数を二つ作って縦と横位置を0から9のどこかに入れるようにしました。でてくる乱数は0から1の間の数だけなので少し変えてあります。それは乱数に10を掛けると $0 \leq 10n < 10$ になり、その整数部分だけを取り出せば求める0から9までの数になります。このサブルーチンは**LBL** 9以降です。

ついでに任意の範囲の乱数を作る方法も説明しておきます。例えば $34.5 \leq x < 98.36$ の乱数を作りたいとします。普通の方法で $0 \leq n < 1$ の乱数を作り63.86(=98.36-34.50)倍すると、0から63.86の範囲の乱数になるので、これに34.5を足せば希望する範囲の乱数になります。

ユーザ指定キー

HP-11Cに**A**、**B**、**C**、**D**、と**E**の5個のユーザ指定キーがあるので非常に便利です。これは次のような3種の使い分けができます。

1. 特定のレジスタにデータをストアする（213ページ参照）。
2. 一つのプログラム中の別々のルーチンを使い分ける。
3. プログラム・メモリーに複数のプログラムを入れておいてそれを使い分ける。

複利計算のプログラムに上の1と2の方法を使っています。

データのストア

複利計算では全部で5個のデータ（1期間の利率 i ，期間数 n ，初回の金額 PV ，毎回の出入金 PMT ，最終回金額 FV ）の内の3～4個をストアする必要があります。うまい具合にユーザ指定キーが5個あるので、それぞれのキーで別々のレジスタにストアできるようにしました。例えば **C** のキーを押すと **f LBL C**，**STO 3**，**R/S** の部分を実行します。そこで **C** を押すたびに X レジスタの値を R_3 にストアします。データが5個までならこの方法でユーザ指定キーに割当てられます。しかもこの方法ならどんな順序でもストアできます。単にデータをキーインしてそれに応じたキーを押すだけです。

別々のルーチンの使い分け

普通のプログラムは複数のデータを使って計算します。どれとどのデータを使って計算するかをユーザ指定キーで使い分けすればよいわけです。それぞれのキーを押せばその計算に必要なデータを選んで計算し答を出します。例えば複利計算のプログラムでは PMT を求めるには次のようにキーを押すようにしました。**D** (PMT を選ぶ)，**R/S** (計算する)。**D** を押すと **LBL D** から始まるルーチンを使うのだという計算機に教えたのと同じです。このプログラムではそのときの数値を R_4 にストアし、それから PMT の計算をします。**R/S** を押す前に表示していたストアした数値が不適当な値でも、 PMT の計算途中には R_4 の値を使わないし、出た答(PMT)を R_4 にストアし直すので不都合はありません。

以上のようにユーザ指定キーは非常に便利で、一つの目的だけでなく各種の使い分けができます。

付 録 A

エラー発生原因

計算途中で不適当な操作（例えば0で割る）をすると Error の文字と数字を表示します。そのときはどのキーを押しても Error の文字などが消えます。

次のようなときに Error の文字などを表示します。

Error 0 : 数学的に不適当な計算

計算に使った数値が不適当です。

$\boxed{\div}$,	$x = 0$ のとき
$\boxed{y^x}$,	$y = 0$ で $x \leq 0$, または $y < 0$ で x が整数でないとき
$\boxed{\sqrt{x}}$,	$x < 0$ のとき
$\boxed{1/x}$,	$x = 0$ のとき
$\boxed{\text{LOG}}$,	$x \leq 0$ のとき
$\boxed{\text{LN}}$,	$x \leq 0$ のとき
$\boxed{\text{SIN}^\circ}$,	$ x > 1$ のとき
$\boxed{\text{COS}^\circ}$,	$ x > 1$ のとき
$\boxed{\text{STO}} \boxed{+}$,	$x = 0$ のとき
$\boxed{\Delta\%}$,	Yレジスタの値が0のとき
$\boxed{\text{HYP}^\circ} \boxed{\text{COS}}$,	$ x < 1$ のとき
$\boxed{\text{HYP}^\circ} \boxed{\text{TAN}}$,	$ x > 1$ のとき
$\boxed{\text{C}y.x}$,	1. x か y が整数でないとき 2. x か y が負数のとき 3. $x > y$ のとき 4. $x \geq 10^{10}$ のとき
$\boxed{\text{P}y.x}$, $\boxed{\text{C}y.x}$	のときと同じ

Error 1: 記憶レジスタのオーバーフロー

$\boxed{\Sigma+}$, $\boxed{\Sigma-}$ 以外の操作で、レジスタ内の数値が $9.99999999 \times 10^{99}$ よりも大きくなったとき。

Error 2: 統計的に不適当な計算

$\boxed{\bar{x}}$	$n = 0$ のとき
\boxed{s}	$n \leq 1$ のとき
$\boxed{\bar{y}_r}$	$n \leq 1$ のとき
$\boxed{\text{L.R.}}$	$n \leq 1$ のとき

注 下記の計算途中で0で割ったり、平方根の中が負数になったときにも Error 2 の表示になります。

$$s_x = \sqrt{\frac{M}{n(n-1)}} \quad s_y = \sqrt{\frac{N}{n(n-1)}} \quad r = \frac{P}{\sqrt{M \cdot N}}$$

$$A = \frac{P}{M} \quad B = \frac{M \Sigma y - P \Sigma x}{n \cdot M} \quad (A \text{ と } B \text{ は } \boxed{\text{L.R.}} \text{ を押し}$$

たときの答で $y = Ax + B$
になります。)

$$\hat{y} = \frac{M \Sigma y + P(n \cdot x - \Sigma x)}{n \cdot M}$$

ただし

$$M = n \Sigma x^2 - (\Sigma x)^2$$

$$N = n \Sigma y^2 - (\Sigma y)^2$$

$$P = n \Sigma xy - \Sigma x \Sigma y$$

Error 3: レジスタ番号が不適當

使おうとしたレジスタがプログラム・メモリーに変わっていたり、もともとない番号を使おうとしたとき。

Error 4: ライン番号やラベルが不適當

そのときにないライン番号を指定した、絶対のないライン番号 (204 以上) を指定した、204 ライン以上のプログラムを入れようとしたとき。またはプログラム中に使ってないラベルを指定したとき。

Error 5: サブルーチンを重ねすぎた

サブルーチンを5段以上にしようとしたとき。

Error 6: フラグ番号が不適當

0 または 1 以外のフラグを使おうとしたとき。

Error 9: 修理が必要

233 ページの動作の確認をご覧ください。

Pr Error

電池が駄目になったなどのために不揮発性メモリーがリセットされたとき。

付 録 B

スタック上昇とラスト X

HP-11C は普通の計算手順通りに操作すればよいように設計してあります。この本でこれまでお読みになったように、自動メモリー・スタックの中がどうなっているかをいちいち考えなくても、筆算でやるときの順序で 1 回に一つの計算を順々に続けられればよいわけです。

しかしあなたが HP-11C 用にプログラムをお作りになるときに、どういう操作をするとスタック内のデータ移動があるかをお知りになっていると便利だと思いますので説明しておきます。

数値入力の区切り

数値をキーインしてからどれかのキーを押したり、プログラムを実行する（走らせる）と数値入力を区切った（打切った）こととなります。つまり計算機はそれ以降にキーインする数値は前のとは全然別の数値だと判断するわけです。（ただし **CHS**、**.**、**EEX** と **+** キーだけは数値入力を区切ったことになりません）

スタック上昇

キー操作をした後で新しい数値をキーインするとそれまでスタック内にあった数値がどうなるかをキー操作別に区別すると次の 3 種になります。つまり次にスタック上昇しない操作、次にスタック上昇する操作、無関係の操作です。

次にスタック上昇しない操作

HP-11C では下の 4 種の操作だけが次にスタック上昇しない操作です：この次にスタック上昇しない操作とは、それぞれのキーを押した後に新しい数値をキーインしても、表示している Xレジスタの値が変わるだけでそれ以外のスタック内の数値が移動することはありません。この 4 種とは次のキーです。 **ENTER** **CL_r** **Σ+** **Σ-**

* 29ページの脚注も見てください。

次にスタック上昇する操作

大部分のキーは ($\boxed{x^2}$) や (\boxed{x}) のように単項演算関数や二項演算関数も含めて) 次にスタック上昇する操作です。これらのキー操作をした後で数値をキーインすると、それまでスタック内にあった数値が一つずつ上昇します。PRGM モードから計算モードに戻したり、計算機のスイッチを切つてからスイッチをもう一度入れる操作も次にスタック上昇する操作です。

T →				
Z →				
Y →		4.0000	4.0000	53.1301
X →	4	4.0000	3	5.0000

キー → 4 **ENTER** 3 **g** **→P**

次にスタック上 スタック上 次にスタック
昇しない操作 昇しない 上昇する操作

T →				
Z →				
Y →	53.1301	53.1301	53.1301	53.1301
X →	0.0000	7	1.0000	9

キー → **g** **CLx** 7 **Σ+** 9

次にスタック上 スタック上 次にスタック上 スタック上
昇しない操作 昇しない 昇しない操作 昇しない

無関係な操作

CHS や **FIX** などは無関係な操作、つまりその操作の前の操作によってスタック上昇するかないが決まる操作です。例えば次にスタック上昇しない操作の **ENTER** を押した後に **f** **FIX** *n* と押し、それから新しい数値をキーインしてもそれまで表示していた X レジスタの値が書替るだけでスタック上昇はしません。また次にスタック上昇する操作 ($\boxed{x^2}$ など) の後に **f** **FIX** *n* と押し、それから新しい数値をキーインするとそれまでスタック内にあった数値が一つずつ上昇します。HP-11C では次の操作が無関係な操作です。

FIX
SCI

GTO **.** *nnn*
BST

CLEAR **PREFIX**
CLEAR **REG**

ENG
DEG
RAD
GRD

SST (計算モードで SST を
押してプログラム命令を
一つ実行すると次にスタ
ック上昇します。)

MEM
USER
P/R

CLEAR Σ
CHS*
PREFIX
R/S
PSE

ラスト x

次の操作をするとそれまでの x はラスト x レジスタに入ります。

-	$\Sigma+$	e^x	\sqrt{x}	RND
+	$\Sigma-$	LOG	x^2	HYP SIN
\times	%	10^x	$1/x$	HYP COS
\div	$\Delta\%$	SIN	y^x	HYP TAN
\rightarrow H.MS	\hat{y}, r	SIN^{-1}	\rightarrow R	HYP ² SIN
\rightarrow H	$x!$	COS	\rightarrow P	HYP ¹ COS
ABS	FRAC	COS^{-1}	$P_{y,x}$	HYP ¹ TAN
\rightarrow R	INT	TAN	$C_{y,x}$	
\rightarrow D	LN	TAN^{-1}		

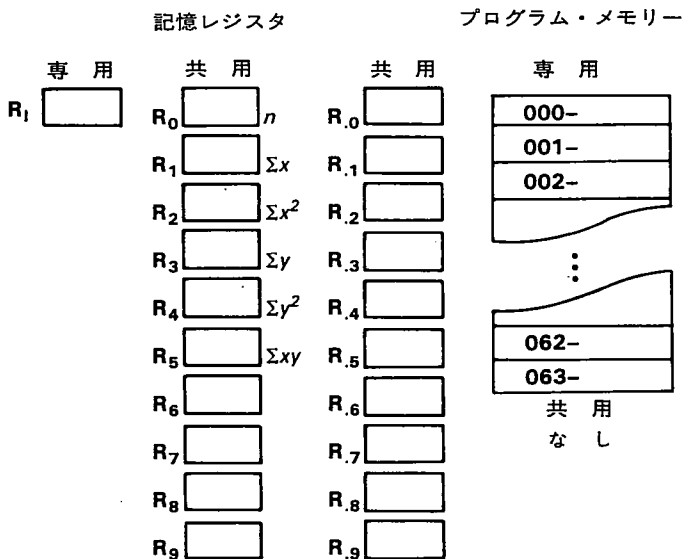
* 数値をキーインしている途中の CHS は無関係な操作です。例えば 1, 2, 3, CHS と押すと -123 をキーインしたことになり、1, 2, 3, EEX, 6, CHS は 123×10^{-6} をキーインしたことになります。CHS を単独で操作すると次にスタック上昇する操作になります。

付 録 C

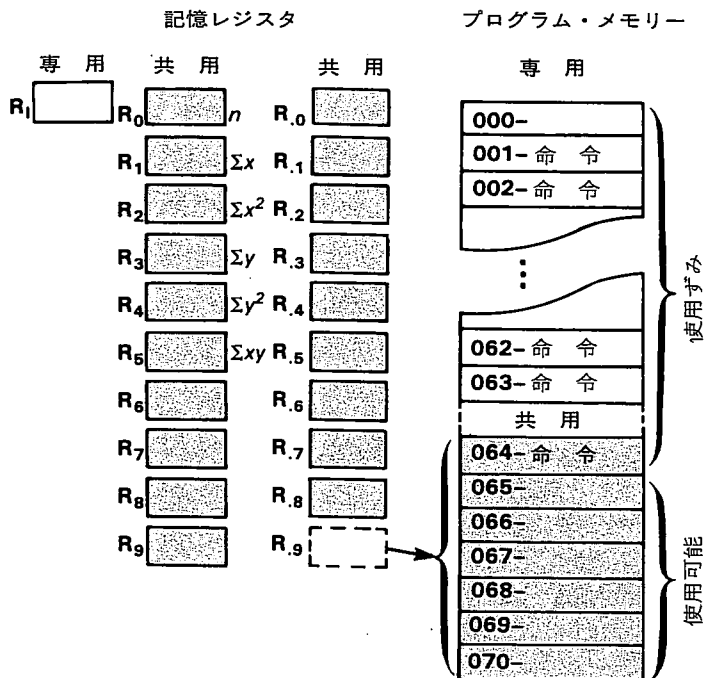
メモリーの自動配分

記憶レジスタからプログラム・メモリーへの変換

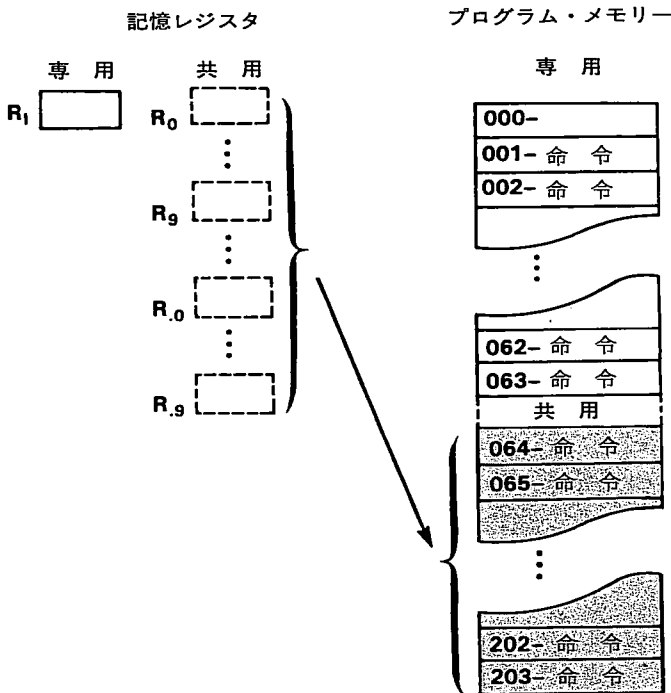
HP-11Cはプログラムに必要なだけ記憶レジスタをプログラム・メモリーに自動的に変換するように設計してあります。プログラムを組み始めたときは、プログラム・メモリーが63ライン分、記憶レジスタが20個（この他に第9章で説明したIレジスタが1個）あります。プログラムが63ライン以内なら下図のような配分です。



64ライン目のプログラム命令をキーインすると、記憶レジスタのR₉がプログラム・メモリーの7ライン分に使われてしまいます。このときのメモリー配分は下図の通りです。



プログラム・メモリーの203ライン分全部にキーインすると、メモリー配分は下図のようになります。



このように始めの21個の記憶レジスタ ($R_0 \sim R_9$, $R_{.0} \sim R_{.9}$, と R_1) が変化しない R_1 の1個だけに減ってしまいました。 $R_0 \sim R_9$ と $R_{.0}$ から $R_{.9}$ のレジスタは1個につき7ライン分のプログラム・レジスタに変わってしまったのです。76ページの表はどのレジスタがプログラムの何ライン分に変わるかを示す表です。

このようにそのときに使えるプログラム・メモリー全部に命令が入ると、次の命令をキーインすると直ちに次の記憶レジスタが7ライン分のプログラム・メモリーに自動的に変化します。例えばプログラムを

70 ライン分キーインしてから 71 ライン目をキーインすると R₈ が 7 ライン分のプログラム・メモリー (71~77 ライン用) に変わり、以下同様に続きます。

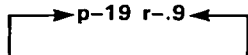
注 HP-11C の記憶レジスタがプログラム用に変化するのには番号の逆の順序 (R₉ から R₀, それから R₉ から R₀) です。そのためあなたがプログラム中で **[STO]** や **[RCL]** するときには番号順 (つまり R₀ からの順) に使うようにしてください。こうすれば使うつもりだった **[STO]** や **[RCL]** の番号がプログラム・メモリーに変わっていて使えなかったということがなくなります。また記憶レジスタに記憶させておいた数値は、そのレジスタがプログラム・メモリー用になると数値が消滅してしまうことも覚えておいてください。

プログラム・メモリーから記憶レジスタへの変換

PRGM モードで **[f] CLEAR [PRGM]** と押すとそれまでプログラム・メモリーになっていた部分 (064 から 203 ラインまで) が記憶レジスタの R₀ から R₉ に戻ります。それだけでなくプログラム命令の一部を削除して、共用のプログラム・メモリー中の命令がなくなると直ちに相当する記憶レジスタに戻ります。

[MEM] の使いかた

[MEM] (memory: メモリー) キーは PRGM モードでも計算モードでもそのときのメモリー配分を知るためのものです。例えば 44 ライン分のプログラムが入っているときに **[9] [MEM]** と押すと次のような表示になります。

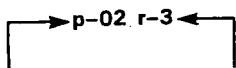


そのときのプログラム・メモリーだけで (記憶レジスタをプログラム用に変えないで) まだ入れられるライン数。

次にプログラム・メモリー用に変わるレジスタの番号。

228 付録C メモリーの自動配分

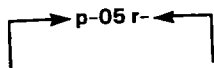
173 ライン分のプログラムが入っているときに **g** **MEM** と押すと次のような表示になります。



そのときのプログラム・メモリーだけで（次の記憶レジスタをプログラム用に変えないで）まだ入れられるライン数。

次にプログラム・メモリー用に変わるレジスタの番号。

198 ライン分のプログラムが入っているときに **g** **MEM** と押すと次のような表示になります。



プログラム・メモリーが一ぱいになるまでにまだ入れられるライン数。

もうプログラム・メモリー用に変わるレジスタがない。

R_i は特殊機能がある永久的な記憶レジスタなので **MEM** を押しても表示しません。

注 統計関係の計算には $R_0 \sim R_5$ のレジスタを使います。この内の一つでもプログラム・メモリーに変わっているときに統計計算をしようすると Error 3 を表示します。

付 録 D

電池、保証と修理について

電 池

HP-11C は電池を 3 個使っています。アルカリ電池でごく普通の使いかたでは 6 ヶ月以上使えます。HP-11C に付属の電池はアルカリ電池ですが、酸化銀電池（寿命が約 2 倍になります）も使用可能です。

新品のアルカリ電池ではプログラムの連続計算（電気を一番多く使っている状態です*）で 80 時間以上使えます。新品の酸化銀電池では同じ状態で 180 時間以上使えます。普通のプログラム計算では連続計算のときよりも電池の消耗が少なくなります。数字などを表示しているだけのとき（キーを押したり、プログラム計算の途中でないとき）には電気を少ししか使わないからです。

計算機のスイッチを切つてあれば不揮発性メモリーの内容が消えないようにほんのごく僅かだけ電流を流すだけなので、新品のアルカリ電池で約 1 年半、新品の酸化銀電池で約 2 年は持ちます。

実際の電池の寿命は計算機をどう使うか、つまりプログラム計算の比率が多いかまたは手操作の比率が多いか、さらにどんな計算に使うのかによってかなり変化します*。

ご 注 意

電池を充電しないでください。電池は高温のところに置かないでください。こうすると電池から液が洩れ出し、このまま計算機中に入れると故障の原因になります。電池を燃えている火の中に投げこまないでください。こうすると爆発することがあります。同様に燃すごみの中に入れてください。

- * HP-11C の電流消費量はそのときの状態、つまりスイッチを切つてあるとき（不揮発性メモリーにだけ電流を流す）、待機状態（表示しているだけ）のとき、動作中（プログラム計算や手計算中、あるいはキーを押した）のときによって変わります。スイッチを入れてあるときには待機中と動作中が混じりあった状態になります。そのため実際の電池の寿命は計算機がどの状態だったかによってかなり大幅に変わります。

HP-11C に付属の電池も下表の電池も充電できません。

HP-11C 用電池は次の通りです。

アルカリ電池	酸化銀電池
LR 44 型	SR 44 型 (G-13 型とも言います)

次の電池はアメリカ製なので上の番号と異なりますが相当品です。

Eveready(UCAR)A 76	Eveready(UCAR) 357
RAY-O-VAC RW 82	RAY-O-VAC RS 76 または RW 42
	Duracell MS 76

電圧低下の症状

電池の電圧が低下すると表示部の左下部分に * 印が点滅します。

アルカリ電池を使用のときは

- * 印が見え始めてから連続で 2 時間以上使用できます。†
- スイッチを切っておけば * 印が見え始めてから約 1 か月間は不揮発性メモリーの内容が消えません。

酸化銀電池を使用のときは

- * 印が見え始めてから連続で 15 分以上使用できます。†
- スイッチを切っておけば * 印が見え始めてから約 1 週間は不揮発性メモリーの内容が消えません。

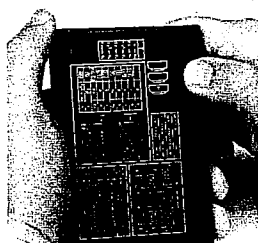
新しい電池との交換

電池交換などで電池を外してもしばらくの間は計算機の不揮発性メモリーの内容は消えません (ただし電池を外す前に必ずスイッチを切ってください)。これは電池交換の間にプログラムやデータが消えないようにするためです。しかしあまり長時間電池を外したままにしておくと不揮発性メモリーの内容が消えてしまいます。

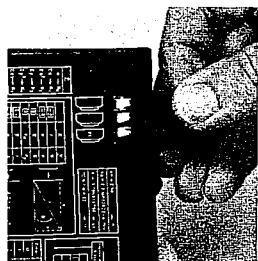
†これは連続してプログラム計算をしているときの最低使用可能時間です。手操作で計算機を使っているときは、動作と待機が組み合わさっている状態なので、* 印が見え始めてからの使用可能時間はこれよりも長くなります。

232 付録 D 電池，保証と修理について
新しい電池と交換するには次のように
してください。

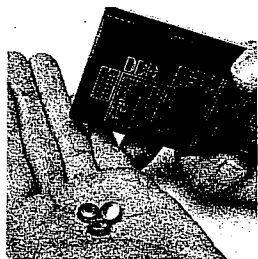
1. 計算機のスイッチを切る
2. 計算機を写真のように持って電池蓋の部分を外側に押しだして少しあけます。
3. 電池蓋の外側をつまんで全部あけます。



注 次の4～5の電池を外している間はキーにさわらないようにしてください。そうしないと不揮発性メモリーの内容が消えてしまうことがあります。



4. 電池側を下に向けて軽く振って、電池を手のひらに落してください。

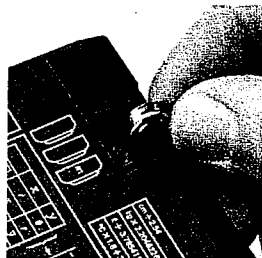


ご 注 意

この次で入れる電池は3個とも新しい電池にしてください。1～2個だけ新しいのにすると、古い電池から液漏れして計算機本体を腐食させてしまうことがあります。

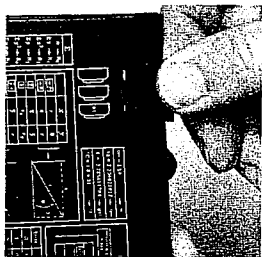
電池の方向（＋と－側）を間違えないでください。間違えると不揮発性メモリーの内容が消えてしまうことがあります。

5. 新しい電池を3個電池室へ入れてください。計算機の裏側の図にあるように平らな面（＋マークがあります）がゴム足の方に向くようにしてください。



6. 電池蓋をつまんで計算機裏側の蓋のみぞに入れます。

7. 電池蓋が裏側と同じ高さになるようにしてから、電池蓋を完全にしめてください。



8. **[ON]**を押してスイッチを入れてください。もし不揮発性メモリーをリセットしてしまったら(内容が消えてしまったら)PrErrorの表示になります。そのときはどのキーを押してもこの表示が消えます。それから必要なプログラムなどを入れてください。



動作の確認

計算機のスイッチを入れても表示しないとか、計算が変なときには次のようにして点検してください。

1. キーを押しても変化がないとか、何も表示しないときにはまず電池をはずしてもう一度入れ直してください。このときに電池の向き（平らで+のマークがある方がゴム足側）に気をつけてください。何回か繰返してもだめでしたら新しい電池を入れてください。それでもスイッチを入れても何の変化がないときは修理が必要です。

234 付録 D 電池、保証と修理について

2. スイッチを切ってから **[x]** キーを押したまま **[ON]** も押してください。
3. **[ON]** を先に離してから **[x]** も離してください。これは計算機内の電気回路の動作試験です。回路が正常なら約 15 秒後（それまでは running の文字を点滅します）に **- 8,8,8,8,8,8,8,8,8,8,8,8** の表示と状態表示の文字を全部表示します*。このときに Error 9 の表示だったり、何も表示しなかったり、何か変な表示のときには修理が必要です。

注 **[+]** キーや **[÷]** キーを押したまま **[ON]** キーを押しても同じように試験できます†‡この 2 種は主として製造工程や修理途中の試験のときにやる方法です。

上記 3 の操作では正しく表示するけれども計算途中で Error 表示をするようでしたら、Error 番号に応じてこの本の該当部分を読み直してください。それでもまだわからないときはお買上げの販売店や YHP に手紙または電話でお問い合わせくださるとよいでしょう。

* この試験をやると HP-11C でいつもは使わないものまで表示します。

† **[ON]** / **[x]** や **[ON]** / **[+]** 試験のあとで Error 9 を表示しても、何とか使いたいときは 20 ページの方法で不揮発性メモリーをリセットしてください。

‡ **[ON]** / **[+]** 試験は **[ON]** / **[x]** 試験と大体同じですが試験終了の表示が出ません。どれかのキーを押すと約 15 秒後に上のような表示になります。**[ON]** / **[÷]** 試験はキーと表示を組合わせた試験です。**[ON]** を離すと表示の一部だけが見えます。試験を始めるには各段ごとに左から右に、第 1 段目が終わったら 2 段目というように全部のキーを順番に押してください。各キーを押すごとに表示が少しずつ変わります。全部のキーを順番通り正しく押すと 11 の表示になるはずですが。（**[ENTER]** キーは 3 段目のときと 4 段目のときの両方に押してください。）計算機が正しく働かないときや、キー順序を間違えると Error 9 を表示します。なおこのようにキーを押す順番を間違えたときに Error 表示が出ますが、この場合には修理の必要はありません。試験を途中で打切るにはわざとキーの順序を間違えてください（当然のことですが Error 9 の表示になります）。Error 9 や 11 の表示はどれかのキーを押せば消えます。

保証について

保証の内容

HP-11C は材質上や製造工程上の不具合に対してお買求めの日から 1 年間の保証をいたします。他の方に贈物としたり、転売された場合は自動的にその人に権利が移りますが、保証期間はあくまでも初めにお買いになりました日から 1 年間だけです。保証期間中の修理・改造・部品交換などの費用とお客様への返送料は当社負担ですが、お客様から当社サービス係への送料はお客様負担でお願いいたします。

保証が適用されない場合

通常使用外での故障（落下、水中に落す等も含む）の場合、当社以外で修理や改造したものは保証の対象外です。また保証期間内でも保証書がないとやはり保証の対象にはなりません。HP-11C の電池は保証の対象外です。

修 理

当社サービス係の住所は下記の通りで直接こちらへお送りください。送料はお客様、返送料は YHP 負担です。

〒229 神奈川県相模原市矢部1-27-15
横河・ヒューレット・パッカード（株）
相模原事業所
パーソナル コンピュータ部
マイクロキャル サービス係
電話 0427-59-1311（大代表）

外国へ出張の場合の修理先は製品に同封のサービスカードをご参照ください。この場合でも保証書は国際的に通用いたします（多少の例外はあるかも知れませんが）のでお持ち下さい。この場合でも修理先への送料、通関手数料、関税等はおお客様の負担になります。

修理料金

前述のように1年間の保証期間がありますが、保証期間切れや期間内でも保証書の添付がないものの修理は有料で機種ごとの定額制になっています。普通の使用法以外の使用による故障修理は別途料金になります。この場合修理前に見積料金を提示いたします。

修理完了後の保証期間

有料修理したものは3ヶ月間の保証（使用部品と修理技術について）をいたします。修理保証書をご覧ください。

修理依頼品の発送について

もし修理が必要になりましたら次の方法でご返送くださるようお願いいたします。

- サービスカードに故障状況を含めて全部記入してください。
- 保証期間内の修理の場合は保証書または修理保証書を付けてください。

輸送途中の破損を防ぐために HP-11C をケースに入れ、サービスカード（または故障状況とお客様の住所、お名前、連絡電話番号を書いた紙）と保証書（保証期間内の場合だけ）を適当な箱に入れて当社サービス係に郵便またはトラック便でお送りください。輸送途中の破損や紛失は保証の対象になりませんので書留小包にしたり、輸送保険を付けることをお勧めします。

現品が保証期間内でも期間満了後でも当社サービス係までの送料はおお客様のご負担でお願いします。

修理が完了しましたら保証期間内のは送料 YHP 負担でお送りいたします。保証適用外のは代金引換郵便にてお送りいたします（大部分の外国も同様です）。お問い合わせは前記サービス係までお願いいたします。

温度範囲

- 使用時 0～55°C
- 非使用時 -40～65°C

受信障害について

HP-11C は弱い電波を出すのでラジオやテレビ受信障害の原因となることがあります。HP-11C はアメリカ FCC の規則 15 条 J 項のクラス B コンピュータの規定に合格していますが、これは一般住宅用受信機を対象にしています。したがって特殊なラジオ/テレビに対しては保証しておりません。もし HP-11C がラジオ/テレビ受信の障害原因と思われましたら HP-11C のスイッチを入・切して確かめてください。もし障害の原因なら次の方法を組合わせて防止してください。

- 受信アンテナやフィーダの方向を変えてみる。
- HP-11C の使用方向を変えてみる。
- 受信機と HP-11C の距離を離してみる。

プログラム手法の索引

これは第3部の各種のプログラムで使ったプログラム手法の索引です。各項目中の太字はページで、細字が該当するライン番号です。

ISG : 142, 002 ; 144, 083 ; 150, 015 ; 151, 081, 086 ; 172, 123.

$x \geq (i)$: 151, 079, 084, 089.

L.R. と関連の関数 : 165, 029, 036, 039, 045, 056 ; 178, 003, 024, 033, 042 ; 183, 023.

サブルーチン : 142-143, 001-009, 010-068, 057-068, 069-078 ; 144, 079-085 ; 150, 020-048, 041-048 ; 151, 064-076, 077-091 ; 156-157, 054-066, 068 ; 160, 046-050, 051-059 ; 171-172, 093-127, 128-131 ; 189, 063-087, 088-092 ; 196-198, 029-040, 054-098, 080-094, 099-123, 133-139.

GTO **I** : 165, 012.

条件判断 : 156, 037 ; 157, 060, 065 ; 160, 055 ; 165, 026 ; 171, 075 ; 188, 010 ; 196, 023, 058 ; 197, 069, 083, 087, 115.

STO **(i)** : 150, 014.

STO **+** n : 160, 016, 018, 024, 026, 049 ; 183, 012, 016 ; 196, 019, 043.

STO **x** n : 156, 019.

DSE : 150, 042 ; 156, 051.

データ入力 : 150, 008-016 ; 156, 001-016 ; 159, 001-003, 004-008 ; 165, 020-030 ; 170-171, 001-006, 022-027, 038-043, 055-060, 087-092 ; 178, 001-004, 023-025 ; 188, 001-003, 013-015, 031-033, 039-041, 052-054 ; 195-196, 001-005, 006-013, 017-022, 041-044.

データ出力 : 150-151, 055-063 ; 165, 038-048 ; 172, 115-124 ; 178, 005-022 ; 179, 081-085.

統計関係 : 165, 029, 039, 045, 056 ; 178, 003, 006, 009, 024, 033, 042, 183, 024.

フラグ : 143, 065, 071, 080 ; 144, 087, 095 ; 164, 010, 011 ; 165, 014, 017, 020, 023, 040, 054, 057 ; 183, 002, 010, 014, 041, 188, 007, 009, 011 ; 189, 072 ; 195, 004, 008 ; 196, 021, 046,

240 プログラム手法の索引

050 ; 197, 107.

RAN# : 195, 009 ; 198, 134.

RCL (i) : 142, 004 ; 144, 080 ; 150, 012, 044 ; 172, 121.

ループ : 150, 008-016 ; 156-157, 026-067 ; 159-160, 009-028 ; 165,
018-030 ; 172, 120-124 ; 178-179, 005-022, 035-085 ; 183, 019
-022, 026-039 ; 188, 004-012.

ユーザ指定キー : 142, 010 ; 143, 079 ; 144, 086 ; 150, 001, 049 ; 156,
001, 017 ; 157, 068, 159, 001 ; 160, 011, 021, 029, 033 ; 164,
001 ; 165, 038, 049, 053 ; 170, 001, 022, 038, 055 ; 171, 087 ;
178, 001, 005, 023, 035 ; 183, 001, 019, 023, 026, 040 ; 188,
001, 013, 031, 039, 052 ; 195, 001, 006 ; 196, 017, 041 ; 197,
124.

キーの索引

[ON] を押すと表示が
ついたり消えたりしま
す (16 ページ)。

一般計算

[\pm][\times][\div] 加減乗除
(22 ページ)。

[\sqrt{x}] 表示している X
レジスタの数値の平方
根を計算する (44 ペ
ージ)。

[x^2] 表示している X レ
ジスタの数値の二乗を
計算する (44 ページ)。

[x!] 表示している X レ
ジスタの数値の階乗ま
たはガンマ関数 $\Gamma(1$
 $+ x)$ を計算する (43
ページ)。

[1/x] 表示している X
レジスタの数値の逆数
を計算する (43 ページ)。

[π] 円周率 (3.141592

654) を X レジスタに入
れて表示する (42 ペ
ージ)。

記憶

[STO] この後に番号 (0
~ 9, .0~.9, **[I]** ま
たは **[i]**) を指定して
X レジスタにある数値
を記憶させる。同様に
レジスタとの直接四則
演算もできます (37 ペ
ージ)。

[RCL] この後に番号 (0
~ 9, .0~.9, **[I]** ま
たは **[i]**) を指定して
そのレジスタにある数
値を X レジスタにコピ
ーする (38 ページ)。

CLEAR [REG] 全部の
記憶レジスタの数値を
0 にする (38 ページ)。

[LSTx] あるキーを押
す前に表示していた数
値をもう一度 X レジ
スタにコピーして表示さ
せる (28 ページ)。

I レジスタ

[I] **[STO]** や **[RCL]**
と組合わせて数値の記
憶に使う。コントロール
数を入れておいて
[STO] や **[RCL]**, 表示,
プログラム中のジャン
プなどの間接指定もで
きる (127 ページ)。

[x \rightleftharpoons i] 表示している
X レジスタの値と R_i
の値を交換する (127
ページ)。

[i] 間接指定用。 **[STO]**
や **[RCL]** と組合せて間
接記憶, 呼出, 四則演算
に使う (130 ページ)。

[x \rightleftharpoons (i)] 表示している
X レジスタの値と R_i
で間接指定したレジ
スタの値を交換する (130
ページ)。

[DSE] I レジスタ内
のカウンタ値を増減値
だけ減少させる。その

減少したカウンタ値が最終値と同じか少くなるとその次のプログラム命令 1 ラインだけを飛越す (128 ページ)。

[ISG] I レジスタ内のカウンタ値を増減値だけ増加させる。その増加したカウンタ値が最終値よりも多くなるとその次のプログラム命令 1 ラインだけを飛越す (128 ページ)。

三角関数

[DEG] 三角関数の角度を度単位にする - G RADかRAD表示が消える (45 ページ)。

[RAD] 三角関数の角度をラジアン単位にする - RAD表示がつく (45 ページ)。

[GRD] 三角関数の角度をグレード単位にする - GRAD表示がつく (45 ページ)。

[SIN] , **[COS]** , **[TAN]** 表示している X レジスタの値 (角度) の三角

関数 (2 辺の長さの比) を計算する (45 ページ)。

[SIN] , **[COS]** , **[TAN]** 表示している X レジスタの値 (三角関数) から逆三角関数 (角度) を計算する (45 ページ)。

順列・組合せ

[P_{y,x}] 順列の計算用。y 個から x 個 (重複なし) 取出して並べるときの並べ方の総数を計算します (52 ページ)。

[C_{y,x}] 組合せの計算用。y 個から x 個 (順番を無視して) 取出す組合せ方の総数を計算します (53 ページ)。

数値入力

[ENTER] 表示している X レジスタの値を Y レジスタにコピーする。二つ以上の数値をキーインするときを使う (27 ページ)。

[CHS] 普通の数値や、**[EEX]** を押した後の数値の符号を変える (17 ページ)。

[EEX] $\dots \times 10^n$ の n を入れる前に押す (71 ページ)。

[0] から **[9]** 数字や番号のキーイン用。

[.] 小数点キーイン用。

数値変換

[RND] X レジスタ内の数値をそのときに表示している通りに四捨五入する (42 ページ)。

[ABS] 表示している X レジスタの数値の一符号を取る (42 ページ)。

[INT] 表示している X レジスタの数値の小数部分を切捨てて整数部分だけにする (42 ページ)。

[FRAC] 表示している X レジスタの数値の整数部分を切捨てて小

数部分だけにする (42 ページ)。

スタック操作

[x↔y] XとYレジスタの数値を入替える (28 ページ)。

[R↓] スタック内の数値を下に一つずつ移動させる (28 ページ)。

[R↑] スタック内の数値を上一つずつ移動 (28 ページ)

[CLx] 表示しているXレジスタの数値を0にする (17 ページ)。

前置キー

[f] キーの向う側に印刷してある黄色の関数や機能を使う前に **[f]** を押す (21ページ)。

[g] キーの手前側の青文字の関数や機能を使う前に **[g]** を押す (21 ページ)。

CLEAR **[PREFIX]**

[f] や **[g]** を押したり、途中まで押した **[f]**

[SCI] や **[g][HYP*]**

などを取消す。それと同時に表示しているXレジスタ内の数値の全桁を表示する (21 ページ)。

双曲線関数

[HYP][SIN] ,

[HYP][COS] ,

[HYP][TAN]

表示しているXレジスタの数値を使ってそれぞれの双曲線関数を計算する (48 ページ)。

[HYP*][SINE] ,

[HYP*][COS] ,

[HYP*][TAN] 表示

しているXレジスタの数値を使ってそれぞれの逆双曲線関数を計算する (48 ページ)。

対数と累乗

[LN] 表示しているXレジスタの数値の自然対数を計算する (48 ページ)。

[e^x] 表示しているXレジスタの逆自然対数、つまり e の x 乗を計算

する (48 ページ)。

[LOG] 表示しているXレジスタの数値の常用対数 (底が 10) を計算する (48 ページ)。

[10^x] 表示しているXレジスタの逆常用対数、つまり 10 の x 乗を計算する (48 ページ)。

[Y^x] Yレジスタの数値のXレジスタの数値乗を計算する (49 ページ)。

統計

CLEAR **[Σ]** 統計用レジスタ (R₀~R₅) とスタックレジスタを0にする (55 ページ)。

[Σ+] XとYレジスタにある統計データをR₀~R₅に集計する (55 ページ)。

[Σ-] XとYレジスタにある統計データをR₀~R₅から削除する (58 ページ)。

[\bar{x}] **[$\Sigma+$]** で集計したデータから x と y の平均を計算する (60 ページ)。

[s] **[$\Sigma+$]** で集計したデータから x と y の標準偏差を計算する (61 ページ)。

[\bar{y}_r] y の推定値と相関係数。 x を使って回帰直線上の y の推定値を計算して X レジスタに入れて表示する。同時に相関係数 (集計した x と y をグラフ上にプロットしたときに回帰直線にどの位うまくあてはまっているかを示す) r を計算して Y レジスタに入れる (64 ページ)。

[L.R.] 回帰直線。 **[$\Sigma+$]** で集計したデータにうまくあてはまる直線 $y = Ax + B$ の係数を計算して、切片 B を X レジスタに、勾配 A を Y レジスタに入れる (63 ページ)。

[RAN#] 前もって **[STO]**
[RAN#] でストアした

乱数の種を使って擬似一様乱数を計算して X レジスタに入れ表示する (54 ページ)

パーセント

[%] パーセント。 Y レジスタの数値の $x\%$ を計算する (49 ページ)。

[$\Delta\%$] 増減パーセント。 Y レジスタの数値から X レジスタの数値に変化したときの増減率をパーセントで表示 (50 ページ)。

表示指定

[FIX] 固定小数点表示の小数点以下の桁数を指定する (67 ページ)。

[SCI] 浮動小数点表示の小数点以下の桁数を指定する (68 ページ)。

[ENG] 工学浮動小数点表示の 2 桁目以降の桁数を指定する (70 ページ)。

全桁表示 **[f]** CLEAR

[PREFIX] と押すと
[PREFIX] キーを押

している間は X レジスタにある数値の有効数字 10 桁を表示する。同時にそれまで押しかけた前置キーを取消す (CLEAR **[PREFIX]** も見てください) (72 ページ)。

変換

[$\rightarrow R$] X と Y レジスタに入れておいた極座標系座標の大きさ r と角度 θ を使って直交座標系の x と y 座標を計算する (52 ページ)。

[$\rightarrow P$] X と Y レジスタに入れておいた直交座標系の x と y 座標値を使って極座標系の r と θ を計算する (51 ページ)。

[$\rightarrow H.MS$] 10 進数を 60 進数 (時, 分, 秒 または 度, 分, 秒) に変換する (46 ページ)。

[$\rightarrow H$] 60 進数を 10 進数に変換する (47 ページ)。

[$\rightarrow RAD$] 度単位の角

度をラジアン単位の角 \rightarrow DEG ラジアン単 度に換算する (47 ページ)。
 度に換算する (47 ページ) の角度を度単位の角 (47 ページ)。

プログラム用キーの索引

P/R PRGM モードと計算モードの切替。**G** **P/R** と押すと PRGM モード (PRGM 表示がつく) と計算モード (PRGM 表示が消える) が切替わる (78 ページ)。

MEM そのときのプログラム・メモリーとレジスタの配分を表示する (77 ページ)。

A B C D E ユーザ指定キーで、プログラム入力時のラベル指定用と、プログラム計算時のスタート用に使う (79 ページ)

USER 計算機を USER モードにしたり、USER モードでなくするのに使う。USER モードにすると \sqrt{x} ,

e^x , 10^x , y^x , $1/x$ は **f** を押さないと働かなくなり、代わりに **A** から **E** まで **f** を押す必要がなくなります (79 ページ)。

0 1 2 3 4 5 6 7 8 9 ラベル番号にも使う。プログラムの各ルーチンの始めに **LBL** の後にこれを押す (79 ページ)。

LBL ラベル。**A** から **E** や 0 から 9 までと一緒に使って、プログラムやサブルーチンの始めの目印にする (79 ページ)。

GTO ジャンプ。**A** から **E** や 0 から 9, **I** と一緒に使う。手操作するとプログラム・

メモリー中の該当するラベルを探してそこで止まる。プログラム計算中にこの命令に出会うとプログラム進行を一たん止め、プログラム・メモリーの該当するラベルを探してジャンプしそこからプログラム進行を再開する (122 ページ)。

GSB サブルーチンへジャンプ。**A** から **E** , 0 から 9, **I** と一緒に使う。手操作するとプログラム・メモリー中の該当するラベルを探してその計算をする。プログラム計算中にこの命令に出会うとプログラム進行を一たん止め、プログラム・メモリー中の該当するラベルを探してジ

ジャンプしそこから
RTN に出会うまで
 計算を続け、**RTN**
 に出会うと **GSB** の
 次の命令に戻って計算
 を続行する (119 ページ)。

GTO \square *nnn* ライン
 番号へジャンプ。プログラム・メモリー中の
nnn ラインにセットする (97 ページ)。

BST バック・ステップ。プログラム・メモリー中を1ラインずつ戻る (97 ページ)。

SST シングル・ステップ。プログラム・メモリー中を1ステップ先に進める (97 ページ)。

← 左向き矢印。
 PRGMモードでは表示している命令を削除し、それ以降の命令が1ラインずつ繰上がる。計算モードでは表示しているXレジスタの数値や数字を削除する (17 ページ)。

CLEAR **PRGM**
 PRGMモードではプログラム・メモリー中の全命令を削除して000ラインにセットする。計算モードでは単に000ラインに戻すだけ (78 ページ)。

PSE ポーズ (一時表示)。プログラムの進行を一たん止めてそのときのXレジスタの値を約1秒間表示し、それからプログラム進行を再開する (79 ページ)。

R/S ラン/ストップ。プログラム・メモリー中のそのときのラインからプログラム計算を始める。プログラム進行中にこれを押すと止まる (79 ページ)。

RTN リターン (戻る)。プログラム進行中にこれに出会うと000ラインまたは **GSB** の次の命令に戻る (79 ライン)。

SF セット・フラグ。

この後に0または1を押し、そのフラグをセットする (111 ページ)。

CF クリア・フラグ。この後に0または1を押し、そのフラグをクリアする (111 ページ)。

F7 フラグ *n* がセットしてあるか? この後に0または1を押し、そのフラグがセットしてあるかどうかの判断をする。そのフラグがセットしてあればその次の命令を実行します。そのフラグがクリアしてあればその次の命令1ラインだけを飛越します (111 ページ)。

$x \leq y$	$x > y$	$x \neq y$
$x = y$	$x < 0$	$x > 0$
$x \neq 0$	$x = 0$	

条件判断。Xレジスタの値と0またはYレジスタの値を比較判断する。その答がイエスならその次の命令を実行します。答がノーならその次の命令1ラインだけを飛越します (110 ページ)。

事項索引

太い数字はまず見ていただきたいページ，細い数字はその次に見ていただきたいページです。

ア行

- Iレジスタ 8, 75, 80, 127 から
 - の数値の減少 128-129
 - の数値の交換 127
 - の数値の増加 128-129
 - のストアとリコール 127
 - の直接操作 127-129
 - のループコントロール数 128
- アルファ (A~E) ラベル 79
- アンダーフロー 18, 93
- イエス/ノーの判断 110-111
- エラー
 - の回復 33
 - のクリア 20
 - サブルーチンの— 121, 221
 - シンプソン法の— 159-162
 - 統計レジスタの— 219-220, 228
 - 不揮発性メモリーの— 20, 220, 233
 - フラグの— 220
 - 丸め (四捨五入) 誤差 72, 155
 - 修理 220, 234, 235-236
- 円の面積計算のプログラム 83 から
- おかしいとき、計算機が 233-234
- オーバーフロー 18, 96, 219
 - $\Sigma+$ の— 57
- ON** 10, 16, 233
- 温度範囲, 計算機の 237

カ行

- χ^2 (カイ二乗) 検定 182-185
- カウンタ, ループの 214
- 仮数 (有効数字) 21, 67, 68, 71, 72

248 事項索引

- 間接指定 127, 130, 211-212
 - サブルーチンの— 131, 136
 - ジャンプの— 130-131, 136
 - 数値交換の— 130, 135, 241
 - ストアとリコールの— 130, 135, 241
 - の表 131
 - ライン番号の— 130, 131, 137
 - レジスタ直接四則計算の— 130, 135
- 缶の容積計算のプログラム 89-90
- キーコード 77-78, 81-83
- キー操作の省略 78, 127
- キーの第一機能 20, 79
- キーの第二機能 20, 79
- 記憶レジスタ 8, 75-77
 - のクリア 38-39
 - の変換 75-79
 - との直接四則演算 39, 241
 - 番号 8, 37-39
 - 統計用— 55-56
- 擬似一様乱数 (乱数参照)
- 期首払・期末払 (複利計算参照)
- 行列式
 - マトリクス計算の— 140-148
 - 連立方程式の— 149
- 曲線のあてはめ 162-168
 - プログラム中のフラグ 215-216
- 金利計算 (複利計算参照)
- クラメル法則 149
- クリア
 - エラー表示の— 19
 - 記憶レジスタの— 38-39
 - スタックの— 55
 - 統計用レジスタの— 55, 58
 - 表示の— 17-18
 - 不揮発性メモリーの— (リセット参照)
 - プログラム・ラインの— 78
 - プログラム・メモリーの— 98
- 計算機内部の数値 67, 71
- 計算時間 53
- 計算中の running 表示 18, 53, 88
- 計算手順 (アルゴリズム) 206-208
- 計算モード 20

桁区切 16
 決定係数 163
 減少 (DSE) 128-129, 214, 241, 243
 Go to (ジャンプ参照)
 故障発見, 計算機の 233-234
 勾配のプログラム例 121 から
 コントロール数 127, 128から

サ行

最上段のキー 79
 最小二乗法 164
 サブルーチン 210-211, 242
 —実行中の RTN 情報 19, 120
 —ジャンプ 119-120, 122 から
 —の段数 120-121
 3 桁ごとの桁区切 16
 三角関数の角度単位 19, 45, 169
 三角形の計算 169-176
 —の数値入力 213
 自己診断, 計算機の 234
 試験
 計算機の回路の— 234
 プログラムの— 96
 指数曲線 162, 166-167
 指数の範囲 72
 指数の符号 68, 70, 71
 下向移動 (プログラム・メモリー中の) 80, 87, 129
 自動メモリー・スタック (スタック参照)
 ジャンプ
 間接— 130-131, 136-137, 241
 サブルーチン— 119-120, 210
 条件— 112-113
 マトリクス計算のサブルーチン— 210
 無条件— 113
 ラベル 112
 ライン (プログラム編集のときの) 97-98, 103
 修理 233, 235-236
 —の発送について 236
 —保証 235-236
 出力 (答の表示) のルーチン 241
 小数点 16

250 事項索引

状態

- 角度単位 45
 - 表示 16
 - f 21
 - g 21
 - PRGM 20, 78, 80, 85, 88
 - USER 79, 88
- 表示形式 67-71
- シングル・ステップ 97, 100-101
 - プログラムの一ずつの実行 99-100
- シンプソン法 159
- 数字のキーイン
 - 新しい数値 17, 24
 - の区切 17, 18, 221
- 数値記憶レジスタ (記憶レジスタ参照)
- 数値の記憶 (ストア) 37
- 数値変換機能 42
- スタック 8, 80
 - 自動メモリー・スタック 26-27
 - 移動操作 27-28
 - 移動なし 26-27
 - 下降 26-27
 - 上昇 26-27, 29, 31, 32, 38, 221
 - 上昇に無関係な操作 221, 222-223
 - のクリア 38, 55, 56
 - 次に—上昇しない操作 21, 29, 38, 221
 - 次に—上昇する操作 29, 221-222
 - $\boxed{\bar{x}}$ と— 60, 61
 - $\boxed{L.R.}$ と— 63
 - \boxed{s} と— 62
 - $\boxed{\Sigma+}$ と— 56
 - \boxed{RCL} $\boxed{\Sigma+}$ と— 57
 - $\boxed{\bar{y}.r}$ と— 65-66
- 正常な動作 (計算機の) 233-234
- 精度 (丸め誤差参照)
- 正負符号の変更 17, 42, 223
 - 指数部分の— 71
- 000 ライン (メモリーのトップ参照)
- 前置キー 21, 78, 79, 127
 - 操作が必要なキー 20-21, 79
 - の取消 21
- 潜水艦撃沈ゲーム 194-205

- 一の乱数発生 217
- 占有しているレジスタ (記憶レジスタ参照)
- 占有しているプログラム・メモリー (プログラム・メモリー参照)
- 専用の記憶レジスタ 8, 76, 224
- 専用のプログラム・メモリー 76, 224-226
- 増加 (ISG) 128-129, 132-134, 214, 241
- 一の限界 134
- 間接指定との組合せ 211-212

タ行

- 台形法 159-162
- 対数曲線 162, 167-168
- 種 (乱数の) 54-55, 194, 217
- 単項演算関数 21, 24, 43-49
- 中間結果 23-25, 26, 31
- 直線のあてはめ 162, 244
- DSEの限界 134
- t検定 176-181
- 停止 (プログラムの) 79, 89-90, 93
- 停止してストア 8, 75-77
- Tレジスタ (定数計算のときの) 34-36
- 適合度の検定 182
- テスト
 - 計算機の回路の一 234
 - プログラムの一 96
- データ入力のルーチン 241
- Δx の限界 214
- 電圧低下の状態 231
- 電気
 - 一使用量 230
 - 一的事故 45, 54, 220
- 電池 230-233
- 統計
 - 集計したデータの訂正 58-59
 - 一計算の精度 56-57
 - 一計算のエラー 219, 228
 - 一用レジスタ 55-56, 224, 228
 - 一用レジスタのクリア 55, 58

ナ行

-
- 長い計算 31-32
 - 二項演算関数 22, 24, 29, 49-54
 - $\Sigma+$ 55
 - ニュートン法 154-158
 - ーのループ 214
 - 入力の順序 22
 - 入力データ (数値) 212-213, 217, 218
 - 熱損失計算のプログラム例 12, 80から

ハ行

-
- 配分 (メモリーの) 8, 75, 76, 77
 - バックステップ 97, 101, 104
 - 番号 (番地)
 - 間接 (間接指定参照)
 - ラベルー 79, 94
 - レジスター 37-39
 - PRGM 表示 20, 78, 80, 85, 88
 - ピタゴラスの定理のプログラム 98から
 - 左向き矢印のキー 17, 29, 98
 - 表示
 - Xレジスタのー 26
 - 状態ー 16
 - FIX と SCI の自動切替 67
 - ー形式 19, 67
 - ー指定 19, 67
 - ーのクリア 17, 19
 - ーの丸め (四捨五入) 67, 68, 69, 70
 - 無表示 233-234
 - 不揮発性メモリー 19, 75, 78, 96
 - 角度単位 45
 - 電池の交換 20, 231-232
 - 表示指定 67
 - ーのエラー 20, 220, 223
 - ーのクリア (リセット参照)
 - ーの内容消滅 231, 232
 - ーのリセット 20, 38, 45, 54, 75, 220, 233
 - フラグ 19, 111
 - 乱数の種 54
 - 複利計算 185-194

符号

指数部分の一 68, 71

数値の一 17, 68

負数 17

フラグ 19, 96, 111, 116-118, 241, 215-216, 239

プログラム

—作成の方針 206-210

—操作キー 245-246

—できない機能 88

—手法の索引 239-240

—の一時停止 91-92

—のエラー 96

—の終り 84

—のキーコード 78

—の構造 206-208

—の実行 87, 115, 118

—の停止 79, 88-91, 93

—の始め 84

—の始めの状態指定 96

—の入力 85-86

—のラベル 94

—変更の例 101-106

—命令の削除 18, 98, 105-107

—命令の追加 102, 105

—命令の番号つけ直し 102

—モード 20

プログラム・メモリー 8, 78, 80-81

そのときの— 84, 87, 100, 105

—中の進行 80, 87, 125

—中の命令の削除 105-107

—の終り 84

—のクリア 78

—の最後 93, 100

—の始め(トップ) 79

—の配分 74-77, 224-227

フローチャート 116, 208-210, 216

べき乗曲線 (累乗曲線参照)

変数の入力 212-213

保証について 235-236

保留しているサブルーチンの RTN 情報 19, 119-120

254 事項索引

マ行

マトリクス計算

- 中の ISG 211
- 中の間接指定 211
- のサブルーチン 210

丸め誤差 (四捨五入誤差) 72, 155, 169

統計計算の— 59

無条件ジャンプ 112

メモリー

記憶レジスタへの変換 76-77, 227-228

—の自動配分 74-77, 224-227

—の変換 74-77, 224-227

MEM 機能 77, 227-228

問題点の把握 206

ヤ行

USER 表示 79, 88

ユーザー指定キー 213, 217-218, 242

ユーザーモード 79, 88

ラ行

ライン番号 77-78

ラスト X 8, 28, 32-34, 80, 223

—を使った定数計算 36

—とクリア 38-39

順列・組合せと— 53

$\Sigma+$ と— 56

統計計算と— 32-33, 65

パーセントと— 50

変化率と— 50

$\bar{y}.r$ と— 65-66

ラベル 79, 84, 94

—を探す 87

—の選択 79

乱数 54-55, 217, 241

リセット (不揮発性メモリーの) 20, 38, 45, 54, 75, 220, 233

累乗曲線 162, 168

ループ 113-114, 214-215, 241

回路電流 147

- コントロール数 128
- 条件判断を使った一のコントロール 114-115
- Δx の限界 214-215
- ニュートン法の条件判断 155
- マトリクス計算の一 148
- 一カウンタ 127, 128から, 132-134, 214
- 一の出口 113-115, 127, 128-130, 132-134, 214-215
- 連続移動 101
- 連立方程式 (マトリクスを使った) 140-148
- 連立方程式 (クラメルの法則を使った) 149-153
- ロールアップ 28
- ロールダウン 28

- 1 まず始めましょう
- 2 自動メモリー・スタック，ラストx，数値のストア
- 3 関数
- 4 表示指定
- 5 プログラム作成の基礎知識
- 6 プログラムの編集
- 7 プログラムの条件判断とコントロール
- 8 サブルーチン
- 9 Iレジスタ
- 10 各種のプログラム
- 11 プログラム作成の技術



横河・ヒューレットパカード株式会社

本社/東部支社：〒168 東京都杉並区高井戸東3-29-21
Tel. 03-331-6111 (大代表)
西 部 支 社：〒532 大阪市淀川区西中島5-4-20 (中央ビル)
Tel. 06-304-6021 (代表)

部 品 番 号
00011-90008 Japanese

Printed in Canada 1/87
Mfg. No. 00011-90071