

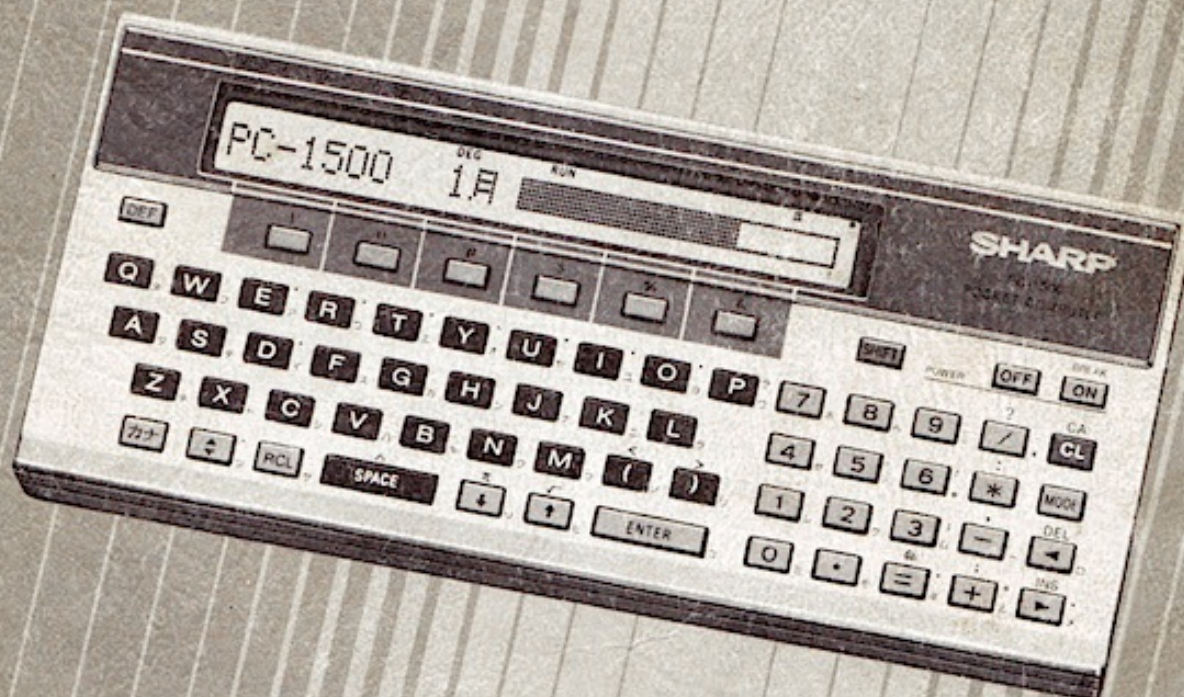
# SHARP

ポケット コンピュータ

形名 **PC-1500**

取扱説明書

保証書付(巻末)  
WITH WARRANTY CARD



**WWW.  
PC-1500  
.INFO**





WWW.  
PC-1500  
.INFO

## 〈はじめに〉

このたびはシャープポケットコンピュータ（PC-1500）をお買いあげいただき、まことにありがとうございました。

この計算機は、プログラム言語にパーソナルコンピュータなどで広く使用されていますBASIC（ベーシック）を用いたプログラム計算機です。

BASIC は世界でもっとも多くの人々が使用しているといわれるプログラム言語で、初めての人でも、少しの学習で容易にプログラムを組むことができます。

**本機を正しくお使いいただくために、この取扱説明書をよくお読みください。**

なお、この取扱説明書は、「シャープサービス・お客様ご相談窓口所在地一覧表」とともに、必ず保存してください。万一ご使用中にわからないことが生じたとき、きっとお役に立ちます。

### （ご注意）

- 本機は非常に複雑な機能および組合せを有する製品であり、出荷に際しては取扱説明書を含めて十分なチェックをして万全を期しておりますが、万一ご使用中ご不審な点・お気づきのことがありましたらもよりのシャープサービス・お客様ご相談窓口までご連絡ください。
- 本書記載のプログラムを使用したことによる金銭上の損害および逸失利益または第三者からのいかなる請求についても当社はその責任を負いませんので、あらかじめご了承ください。

## 目 次

(ページ)	(ページ)
●おねがい..... 4	5-2) 配列変数(DIM).....42
●本機的主要な特長..... 5	5-3) メモリーの構成と変数..... 44
●電池の入れかた、交換のしかた..... 6	5-4) 変数(メモリー)の内容 を消すには(CLEAR).....46
1. キー操作と表示..... 8	6. 表示に関するBASIC.....47
1-1) 各部のなまえ..... 8	6-1) いろいろな表示のさせかた.....47 (PRINT, USING, CURSOR, WAIT, PAUSE, CLS)
1-2) キー操作のしかた..... 8	6-2) 文字や記号の作りかた.....51 (GCURSOR, GPRINT, POINT)
1-3) モードについて.....10	7. そのほかのBASIC .....55
2. マニュアル計算.....11	7-1) READとDATA, RESTORE .....55
2-1) 基本計算.....12	7-2) ON GOTO, ONGOSUB .....56
2-2) 関数計算.....15	7-3) ON ERROR(エラー処理) .....57
2-3) メモリー計算.....18	7-4) REM.....57
3. プログラム(その1).....20	7-5) ARUN.....58
3-1) 簡単なプログラム.....21 (INPUT, LET, PRINT, END)	7-6) STOP.....58
3-2) プログラムの入れかたと編集.....23 (NEW, LIST)	7-7) AREAD.....59
3-3) プログラムの実行.....25 (RUN)	8. 特別な機能.....60
3-4) いろいろな入力形.....26 (INPUT)	8-1) 乱数関数.....60
3-5) いろいろな出力形.....27 (PRINT)	8-2) TIME.....61
4. プログラム(その2).....29	8-3) INKEY\$.....61
4-1) ジャンプと判断.....29 (GOTO, IF...THEN)	8-4) 音発生機能(BEEP).....62
(1) ジャンプ.....29	8-5) CONT.....63
(2) 判 断.....30	8-6) STATUS, MEM.....63
4-2) くり返し計算.....35 (FOR...TO...STEP, NEXT)	8-7) 論理積、論理和、否定.....64 (AND, OR, NOT)
4-3) サブルーチン.....37 (GOSUB, RETURN)	8-8) 条件式の結合.....66
4-4) 定義付けプログラム.....39	8-9) モードロック .....67 (LOCK, UNLOCK)
5. 変 数.....42	9. デバッグ.....68 (TR ON, TR OFF)
5-1) 変数の構成.....42	10. 文字の編集.....70
	10-1) 文字列の分断.....70 (LEFT\$, RIGHT\$, MID\$)
	10-2) 文字列の結合.....72
	10-3) 文字数を数える方法(LEN).....72



	(ページ)		(ページ)
10-4) 数値と文字の変換.....	73	(2) テープへの記録方法.....	105
(VAL, STR\$)		(3) 計算機とテープの内容の照合.....	107
10-5) アスキーコードの変換.....	73	(4) テープからの転送.....	108
(ASC, CHR\$)		(5) CE-152以外のテープレコーダーを 使用する場合について.....	110
10-6) 文字列の比較.....	74		
11. リザーブ機能.....	76	◆キーの機能.....	111
11-1) リザーブのしかた.....	76	◆オプション機器のご案内.....	116
11-2) ソフトキーの キーシンボル表示.....	79	◆仕 様.....	117
◆カラーグラフィックプリンタについて.....	82	プログラム事例集.....	119
〈CE-150と計算機との接続〉.....	82	エラーコードとエラー内容.....	167
●プリンタに関する命令.....	84	キャラクタ・コード表.....	170
(1) モード指定命令.....	84	関数・命令(省略形)一覧.....	171
(TEXT, GRAPH)		索 引.....	181
(2) プリント命令(LPRINT).....	84		
(3) 文字指定命令.....	85		
(CSIZE, ROTATE, COLOR)			
(4) 色やインキのチェック(TEST).....	86		
(5) 印字位置の指定(TAB).....	86		
(6) 紙送り命令(LF).....	87		
(7) ペン移動命令(GLCURSOR).....	88		
(8) 原点指定命令(SORGN).....	88		
(9) 描線命令(LINE, RLINE).....	89		
(10) リストのしかた(LLIST).....	92		
〈テープレコーダーとの接続方法〉.....	94		
●テープレコーダーに関する命令.....	96		
(1) プログラムの記録命令.....	96		
(CSAVE)			
(2) プログラムの転送命令.....	97		
(CLOAD, MERGE, CHAIN)			
(3) 照合命令(CLOAD?).....	101		
(4) データの記録命令(PRINT #).....	102		
(5) データの転送命令(INPUT #).....	102		
(6) リモートコントロール機能について.....	103		
(RMT OFF, RMT ON)			
◆計算機とテープレコーダーの操作方法.....	104		
(1) 準 備.....	104		

付属のネームラベルにお名前  
をご記入の上、計算機の裏面  
にはってご使用ください。

## おねがい

- 本機は精密部品で構成されていますので、直射日光の当たる場所（特に車の中では非常に温度が高くなる場合があります）や、暖房器具の近くなど、高温になることがある場所に置かないでください。

また、急激な温度の変化、低温、湿気、ホコリ、曲げ、ヒネリ、強いショックなどはさけておつかいください。故障などの原因になるおそれがあります。

- 液晶表示部はガラスでできていますので、強く押さえないでください。割れることがあります。
- お手入れの際は乾いた柔らかい布を使用し、ベンジン、シンナーなどの揮発性の液体や、ぬれぞうきんなどを使用しないでください。

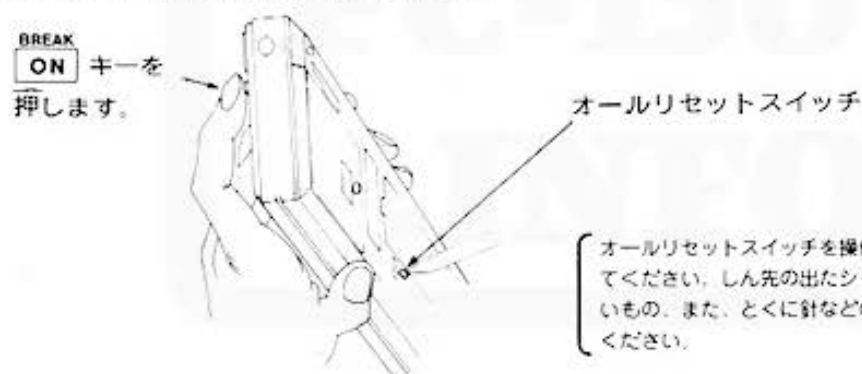
揮発性の液体やぬれぞうきんなどは計算機をいためることがあります。

- 電池を火中に投入しないでください。火中に投入すると破裂するおそれがあります。
- この計算機を分解しないでください。万一故障のときは、お買上げの販売店あるいは「シャープサービス・お客様ご相談窓口所在地一覧表」に記載のもよりのサービス会社へ本書をそえてご持参ください。

## 異常が発生した場合の処理について

本機をご使用中に強度の外来ノイズや強いショックをうけた場合などに、ごくまれに **BREAK ON** キーも含めた、全てのキーが働かなくなるなどの異常が発生することがあります。

本機にこのような異常が発生した場合は、**BREAK ON** キーを押さえたままで計算機裏面のオールリセットスイッチを約15秒間押してください。



そののち、表示部に **NEW 0 ? : CHECK** と表示されることを確認して

**CL** **NEW 0** **ENTER** (0は数字のゼロを示します。11ページ参照)

と押して、計算機内をクリア（消去）し、続いて

**SHIFT** **MODE** **NEW** **ENTER**

と押してリザーブメモリーの内容も消去してください。（リザーブについては76ページを参照してください。）

なお、オールリセットスイッチを押したのち、もしも **NEW 0 ? : CHECK** と表示されなかった場合は、もう一度上記方法でオールリセットスイッチを押してください。

（注）オールリセットスイッチを押しますと、プログラム、データ、リザーブ内容およびTIMEの内容（61ページ参照）が壊れますので、異常状態が発生したとき以外は、このスイッチを押さないでください。



## 本機の主な特長

### ★自由にシンボルを描くミニグラフィック表示

縦7ドット、横156ドットのマトリックス液晶表示部には、ミニグラフィック機能により自由にシンボルを表示させることができます。

ゲーム用シンボルやカタカナ、ひらがな、簡単な漢字など、目的に応じてプログラミングすれば楽しい、そしてスマートな表示をおこなわせることができます。しかも、この表示部は文字・数値表示用としてはワイドな26桁を表示させることができます。

### ★大きなメモリー容量、しかもメモリー保護機能つき

内部にはROM 16Kバイト、RAM 3.5Kバイトを標準実装。さらに別売のメモリーモジュールを取りつけることにより、RAM 4Kバイトおよび8Kバイトの拡張ができます。

この大きなメモリー容量により、まさにパソコンに迫るアプリケーションがポケットコンピュータで可能になりました。しかもプログラムやデータは電源を切っても保護されていますので使うたびに記憶させる必要がなく、いつでも、どこでも、すぐに使用できます。

ROM : Read Only Memory 読み出し専用メモリー

RAM : Random Access Memory 読み書きができるメモリー

バイト : メモリーの大きさやプログラムの大きさを表わす場合などに用いる単位

### ★高速CPU採用で処理時間を大幅に短縮

当社独自開発による高速CMOS CPUの採用により、処理速度が一挙に約10倍（当社比）と高速化に成功！

これまで多くの時間を費やしていた複雑な技術計算、事務計算なども処理時間が大幅に短縮され、効率的に使用することができます。

CPU : Central Processing Unit 中央処理装置

### ★自由に設計できるソフトウェアキー

6個のソフトウェアキー（ソフトキー）は表示と対で使用し、キー操作をリザーブするリザーブキーとして、あるいはプログラムの頭出しができる定義付けキーとして、さらには表示に対してのアクションをおこなうゲーム用キーとして、プログラムにより自由に設計し、使用することができます。

しかも3段切換式になっており、合計18個のソフトウェアキーとして使用することができます。（76ページ参照）

## 電池の入れかた、交換のしかた

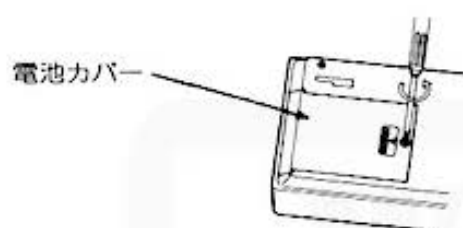
### 電池の交換時期

正面から見て、表示部右端にあるバッテリーインジケータ「●」がうっすらとしか見えなくなった場合は電池の消耗を示していますので、速やかに新しい電池と交換してください。

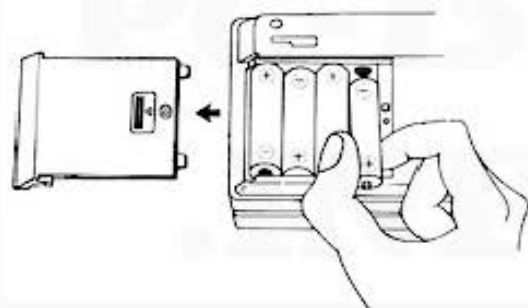
(別売の周辺機器CE-150とCE-152をお持ちの場合は、事前にプログラムやデータをテープに記録してください。)

電池を入れる場合、および交換する場合は次の手順にしたがっておこなってください。

- ① **OFF** キーを押して電源を切ってください。
- ② 硬貨や小型のネジ回して電池カバーのビス(ネジ)をはずし、電池カバーをはずしてください。



- ③ 使用済みの電池を取りはずし、単3タイプの新しい乾電池を4本、電池の⊕・⊖をまちがえないように入れてください。



- ④ 電池カバーをはめてビスで留めてください。(ビスはゆるまないように締めてください。)
- ⑤ ( **ON** ) **CL** NEW 0 **ENTER**

と押して計算機内をクリア(消去)したのち、次のようにプロンプト記号が表示されることを確認してください。



↑プロンプト記号が表示されることを確認してください。

もし、上記以外の表示がなされているときは、電池を取りはずし、改めて入れたのち、4ページの方法でオールリセットスイッチを押してください。



(注) 表示部に **NEW 0 ? : CHECK** と表示されたときは、必ず

**CL** NEW 0 **ENTER**

と操作してください。

もし、上記操作をおこなわなかった場合、メモリー内や計算機の状態が一定せず、プログラムが書き込めないことや、計算機が迷走して止まらなくなることがありますので、必ず操作してください。

もし、迷走して止まらなくなった場合は4ページの方法でオールリセットスイッチを押してください。

なお、上記操作をおこなっても、リザーブメモリーの内容は消去されず、不確定な内容が残る場合がありますので、76ページの「準備」で説明の方法で内容を消去してください。

◆この計算機には単3タイプの乾電池を4本ご使用ください。

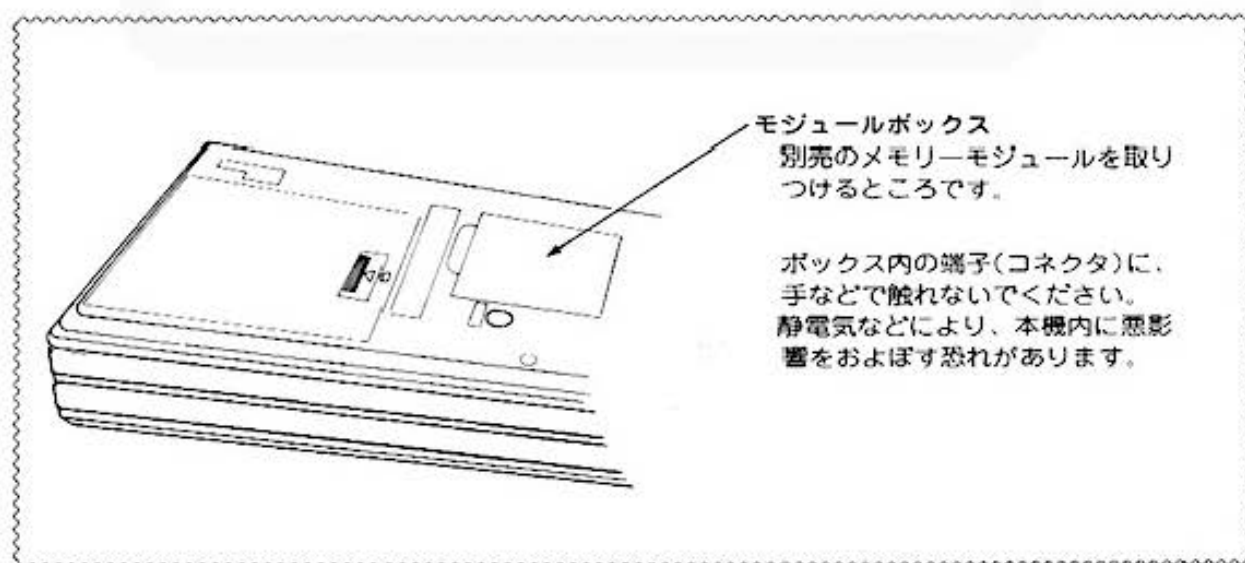
#### 〈乾電池取扱上のご注意〉

電池は誤った使い方をしますと、液もれや、破れつすることがありますので、次の点について特にご注意ください。

- ① 電池のプラス⊕とマイナス⊖の向きを本機の表示通り正しく入れてください。
- ② 新しい電池と、一度使用した電池を混ぜて使用しないでください。
- ③ 種類のちがう電池を混ぜて使用しないでください。同じ形状でも電圧の異なるものがあります。
- ④ 電池は使えなくなったり、長時間使わないときは本機から取り外しておいてください。
- ⑤ 電池には充電式と充電式でないものがあります。電池の注意表示をよく見てご使用ください。

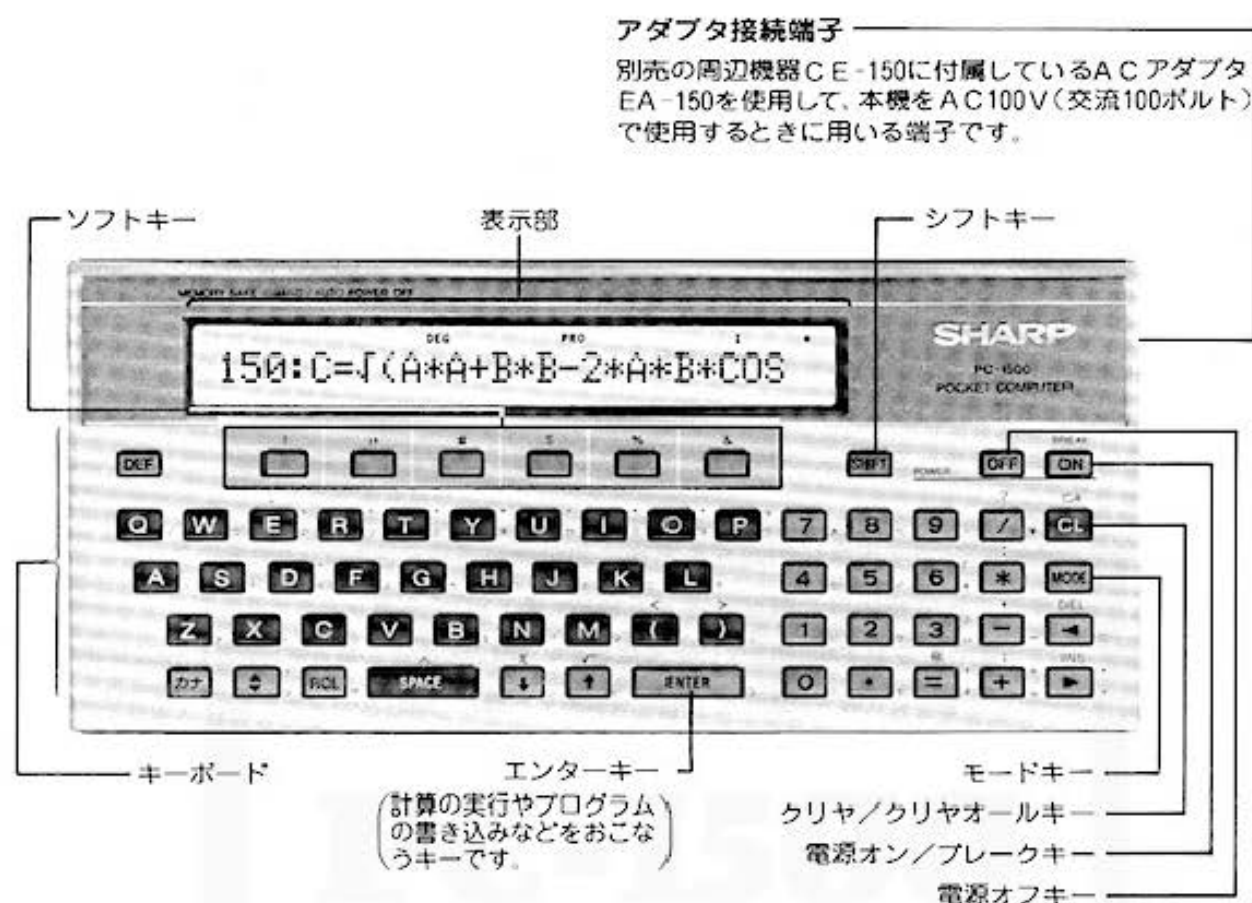
#### 〈ご注意〉

- 消耗した電池をそのままにしておきますと、電池の液がもれて計算機をいためることがあります。
- 電池を火中に投入しないでください。火中に投入すると破裂するおそれがあります。
- 本機には充電式電池は使用できません。(また、充電もできません。)



# 1. キー操作と表示

## 1-1) 各部のなまえ



## 1-2) キー操作のしかた

本機はキーがたくさんあります。(65キー)

各キーはそのまま押す場合と、**SHIFT** キー(シフトキー)に続いて押す場合とがあり、それぞれの場合で働く機能が違います。

ここではキーの操作について簡単に説明します。

### (1) シフトキーに続いて押した場合とそうでない場合の働きの違い

(例) **SHIFT** キーに続いて押せばこちらの機能が働きます。

**DEL** ←そのまま押せばこちらの機能が働きます。

**SHIFT** キーに続いて押せば小文字が入力されます。

**A** ←そのまま押せば大文字が入力されます。

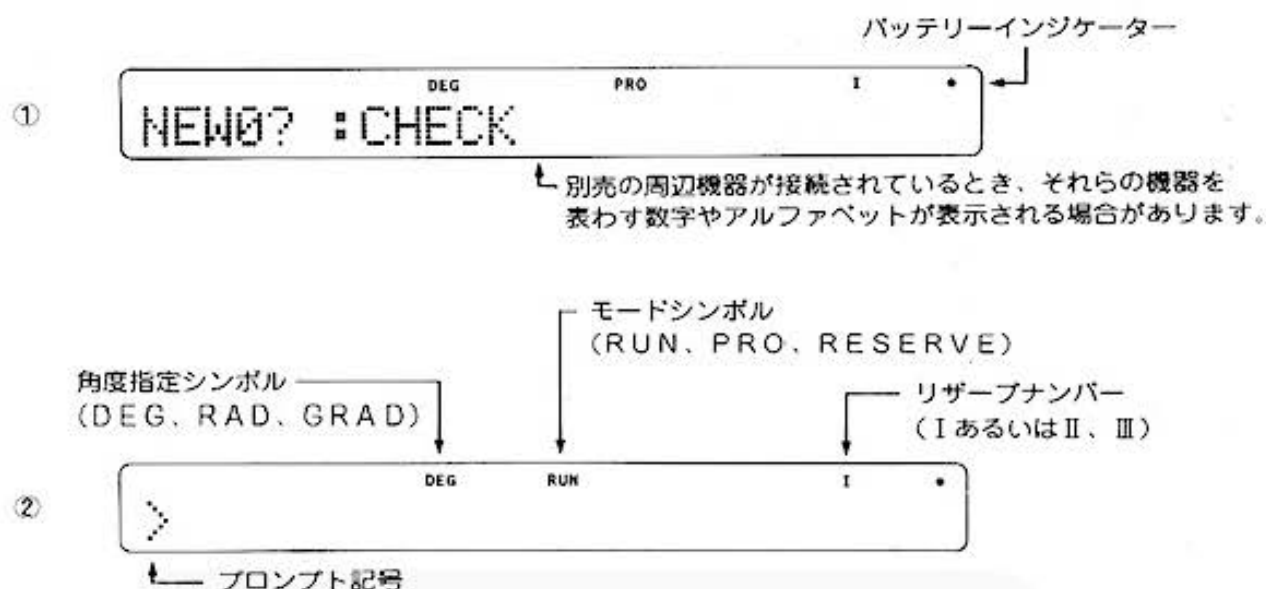
**SHIFT** キーに続いて押せばキーの上側に記されている記号が入力されます。

**+** ←そのまま押せばキーに直接記されている記号が入力されます。



## ② 電源のオン/オフ

まず始めにキーボード右上にある **ON** キーを押してください。電源が入り、次のような表示になります。



①の表示は電池を交換したときなどに表示されます。もし、①の表示になったときはまず最初に次のように操作して計算機内をクリアしてください。

**CL** **NEW 0** **ENTER**

**NEW 0** は“すべての状態を解除(初期状態に)しなさい”という命令です。

この操作をおこなえば、表示は②のように、プロンプト記号が表示されます。

**OFF** キーで電源を切ったのち、**ON** キーで電源を入れれば、②の表示になります。

(ただし、別売の周辺機器が接続されているとき、その周辺機器になんらかの異常がある場合は、たとえば :CHECK 6 のようにチェックメッセージが表示されます。)

## ●オートパワーオフ (AUTO POWER OFF)

本機はキー入力が可能なときに、約7分間新たなキー操作をおこないませんと、電池の消耗を少なくするため、自動的に電源が切れます。

この場合、**ON** キーを押せば電源が入りますが、計算機の状態、表示は電源が切れる前の状態にもどります。

### テンプレート

INPUT	PRINT	USING	GOTO	GOSUB	RETURN	CSAVE	CLOAD	MERGE	LIST
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

テンプレートは計算機のキーボードのアルファベットキー部分に載せて使用します。アルファベットキーは定義付けキーとしてプログラムなどを定義付けることができますが、各キーに定義付けしたプログラムを表わす記号や文字をテンプレートに書いて利用してください。(定義付けプログラムについては39ページを参照)

### (3) アルファベットや記号、数字の入力

電源を入れたのち、**A** ~ **Z**、**0** ~ **9**、**.**、**/**、**\***、**-**、**+**、**=**、**(**、**)**などのキーを押せば、それぞれのキーに直接記されている文字や記号、数字が入力されます。

(例) **CL** **A** **B** **C** → **A B C \_**

**CL** **1** **2** **3** **.** **4** → **1 2 3 . 4 \_**

**CL** **A** **=** **2** **+** **5** → **A = 2 + 5 \_**

**CL** は表示を消去するときやエラーを解除するときに押します。

これはカーソルです。  
次になにか入力すると  
この場所に入ります。

もし、アルファベットの小文字や、キーの上側に記されている記号を入力したいときは、それぞれのキーの前に **SHIFT** キーを押してください。

(例) **CL** **SHIFT** **A** **SHIFT** **B** **SHIFT** **C** → **a b c \_**

**CL** **SHIFT** **/** **SHIFT** **+** **SHIFT** **=** → **? : # \_**

(注) 本機のキーの大部分は横にカタカナや記号が記されていますが、これらの文字や記号および **カナ** キーは別売の「カタカナ」があるときにのみ入力することができます。

## 1-3) モードについて

電源を入れてキーボードの右端の列にある **MODE** キーを何回か押してみてください。

表示部の上のほうで **RUN** と **PRO** というシンボルが交互に表示されることがわかります。

次に **SHIFT** **MODE** と押してください。今度は **RUN** や **PRO** が消えて **RESERVE** が表示されます。(もし、表示が替わらないときは、**UNLOCK** **ENTER** と操作してから押してください。)

これらのシンボルは計算機がどのような状態(モードと呼びます)になっているかを示すシンボルです。すなわち、**RUN** が表示されているときは、計算機が計算を実行できる状態になっていることを示し(この状態をランモードと呼びます)、マニュアル計算や、プログラムを実行します。**PRO** が表示されているときはプログラムを書き込んだり、編集したりすることのできる状態になっていることを示します。(この状態をプログラムモードと呼びます。)

**RESERVE** が表示されている状態をリザーブモードと呼びます。

リザーブモードは、ソフトウェアキーにキー操作などをリザーブするモードです。

(リザーブについては76ページで説明)

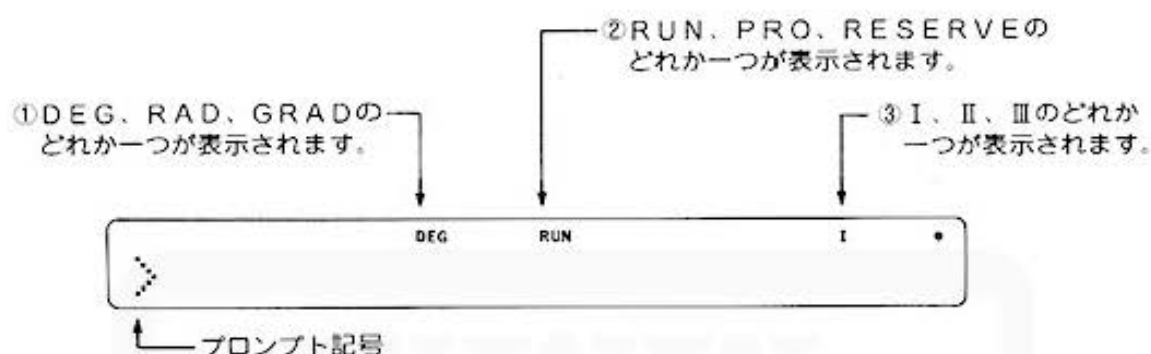


## 2. マニュアル計算

ここでは、計算をマニュアルでおこない、本機の操作になれていただくことと、いろいろな基本計算をマスターしていただくことを目的とします。

### 準備1. 電源オン

まず **[ON]** キーを押してください。電源が入り、表示部の左端に **>** (プロンプト記号) が表示されます。



### 準備2. モード指定

**[MODE]** キーを操作しますと、**RUN**と**PRO**が交互に表示されますので、マニュアル計算のときは**RUN**を表示させてください。(ランモードを指定してください。)

もし、**[MODE]** キーを押しても**RUN**と**PRO**の表示が替わらないときは、**UNLOCK** **[ENTER]** と操作してから、**[MODE]** キーを押してください。

### ＝注＝

本書ではキー操作やプログラムの説明において、数字ゼロ(0)を **①** と表わしている場合があります。①は表示あるいは印字において、アルファベットのオー(O)と区別するための数字で、本書においても、オーとゼロの区別が付きにくい場合はゼロを **①** と表わしています。

## 2-1) 基本計算

〔例題〕次の計算をおこなってください。

- (1)  $2 + 3 \times 4 =$
- (2)  $5 \times (-6) =$
- (3)  $5 \times (-6) + 7 =$
- (4)  $36 \div (1 + 2) =$
- (5)  $\frac{4 + 5}{2 + 3} =$
- (6)  $1 + 2(-3 + 4(5 + 6(7 - 8))) =$
- (7)  $5 \times 10^3 \div (4 \times 10^{-3}) =$
- (8)  $-1.2 \times (-4) + (-5) \times 6 =$
- (9)  $2\pi \times 5 =$

〔解答〕

- |  |             |
|--|-------------|
| (1) $2 + 3 \times 4$ <b>ENTER</b>                                      | 14          |
| (2) $5 \times -6$ <b>ENTER</b>   | -30         |
| (3) $5 \times -6 + 7$ <b>ENTER</b>                                     | -23         |
| (4) $36 \div (1 + 2)$ <b>ENTER</b>                                     | 12          |
| (5) $(4 + 5) \div (2 + 3)$ <b>ENTER</b>                                | 1.8         |
| (6) $1 + 2 \times (-3 + 4 \times (5 + 6 \times (7 - 8)))$ <b>ENTER</b> | -13         |
| (7) $5 \text{ E } 3 \div 4 \text{ E } -3$ <b>ENTER</b>                 | 1250000     |
| (8) <b>CL</b> $-1.2 \times -4 + -5 \times 6$ <b>ENTER</b>              | -25.2       |
| (9) $2 \times \pi \times 5$ <b>ENTER</b>                               | 31.41592654 |
- ↑ **SHIFT**  $\frac{\pi}{\downarrow}$  と押します。

〔説明〕

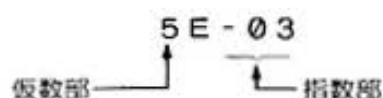
### ◎ 基本キー

通常の数学では四則計算の記号として「+、-、×、÷」を使いますが、本機では「×や÷」は使わず、「\*と/」を使います。これらを次にまとめます。

加える（加算）	+
引く（減算）	-
かける（乗算）	*（アスタリスク）
割る（除算）	/（スラッシュ）
符 号	- または +
指 数※	E（例題の(7)を参照）
計算の実行	<b>ENTER</b>

※20000を $2 \times 10^4$ 、0.005を $5 \times 10^{-3}$ のように表わす方法を浮動小数点表示(指数表示)と呼びます。

これらを本機で表わす場合は指数部を示すEを入れて2E 04、5E-03のような形で表示します。また、キー入力する場合は2E4、5E-3のようにキーを押します。



なお、指数部に3桁以上の数値を入れた場合は、後2桁の数字が有効になります。

(例) 5E123 [ENTER] → 5E 23

### ◎ 計算の優先順位

本機ではカッコ、関数も含めて数式通りのキー操作で計算をおこなうことができます。計算の優先順位の判断や途中結果の処理はすべて計算機が自動的に処理してくれます。計算の優先順位は次の通りです。

1. 関数 (sin, cos など)
2. べき乗 (^) ((例)  $5^3=5^3$ 、 $3^{-2}=3^{-2}$ )
3. 符号
4. 乗除算
5. 加減算
6. 論理演算 (>, >= など)
7. AND, OR

注) ● カッコが使用されている場合はカッコ内の計算が最優先されます。

● 複合関数 (sin cos<sup>-1</sup> 0.6 など) は右から左の順で計算されます。

● べき乗の連算 ( $3^4^2$  即ち  $3^4 \wedge 2$  など) は右から左の順で計算されます。

● 2. と 3. では後に出てきたほうが優先順位が高くなります。

(例)  $-2^4 \rightarrow -(2^4)$

$3^{-2} \rightarrow 3^{-2}$

### ◎ 数式の長さ

一つの数式の長さ ([ENTER] を押すまで) は80キー操作まで可能です。

### ◎ キー操作の訂正

もしも誤ってキー操作をしてもあわてずに次の方法で訂正しましょう。

- (1) 誤りが多いとき

[CL] キーを押せば表示が消えて、左端に「>」記号が表示されますので、最初から入れ直します。

- (2) 誤って他のキーを押したとき

誤って押したキーのシンボルが、点滅表示されるまで [←] キーを押して、そこで正しいキーを押します。( [→] や [←] は押し続けるとカーソル\*が続けて移動します。)

(例)  $2+3*4$  を  $2+3/4$  とした場合は

$2+3\cancel{*}4$

↑ この部分を点滅させて \* を押し、 [ENTER] キーを押します。



## ※カーソル

キー操作をおこなうとき  $\_$  を表示したり、キー操作の訂正のときに点滅表示をしたりしますが、これらは次にキー操作をしたキーがこの部分に入ることを示しており、これをカーソルと呼びます。

## (3) 誤って余分なキーを押したとき

余分なキーのシンボルが点滅表示されるまで  $\leftarrow$  キーを押し、 $\boxed{\text{SHIFT}} \boxed{\text{DEL}}$  と押します。(DEL: DELETE(デリート)…削除を意味します。)

(例)  $2+3*4$  を  $2+3*/4$  とした場合は

$\leftarrow$  を操作  $2+3*/4$   
 $\uparrow$  この部分を点滅させてください。

$\boxed{\text{SHIFT}} \boxed{\text{DEL}}$  と押す  $2+3*4$   
 $\uparrow$  ここが点滅します。

$\boxed{\text{ENTER}}$  を押す  $14$  (答)

## (4) キーの操作がぬけたときは

ぬけたキー操作部分の次のキーシンボルが点滅するまで  $\leftarrow$  キーを押し、そこで  $\boxed{\text{SHIFT}} \boxed{\text{INS}}$  と押すと一文字分の空欄ができますので、次にぬけていたキーを押せば、その位置に入力されます。(INS: INSERT(インサート)…挿入を意味します。)

(例)  $2+3*4$  を  $2+34$  とした場合は

$\leftarrow$  を操作  $2+34$   
 $\uparrow$  この部分を点滅させてください。

$\boxed{\text{SHIFT}} \boxed{\text{INS}}$  と押す  $2+3 \square 4$   
 $\uparrow$  ここが点滅します。

$*$  を押す  $2+3*4$   
 $\uparrow$  ここが点滅します。

$\boxed{\text{ENTER}}$  を押す  $14$  (答)

## ◎ ブレイバック機能

$\boxed{\text{ENTER}}$  キーで計算を実行したのち、 $\leftarrow$  あるいは  $\rightarrow$  キーを押すと実行した式が呼びもとされます。これは実行した式の確認や実行した式の一部を変更して再度実行したいときに便利です。特に  $\boxed{\text{ENTER}}$  で計算を実行した結果、エラーになったときにこのブレイバックを使いますと、エラーを起こした場所をカーソルで示していますので、エラーの原因がわかりやすく大変便利です。(＞記号(プロンプト記号)が表示されたときには、すでに内容が消されていますのでブレイバックできません。)

(例)  $36/1+2$   $\boxed{\text{ENTER}}$  と押してエラーになったとき

$\leftarrow$  を押すと  $36/1+2 \text{ )}$   
 $\uparrow$  点滅表示

## ◎ 連続計算のしかた

**ENTER** キーを押しますとその計算結果が表示されますが、その結果を続けて計算することもできます。

(例)  $1+2$  **ENTER** 3  
 $*4$  **ENTER** 12 (3×4を実行)

## ◎ エラー表示

マニュアル計算でエラーになりますと

**ERROR 1** や **ERROR 37**

のようにエラー番号が表示されます。(エラー番号の意味は167ページの「エラーコードとエラー内容」を参照ください。)

## 2-2) 関数計算

[例 題] 次の計算をおこなってください。

- (1)  $\sin 30^\circ + \cos 60^\circ \times \tan 45^\circ =$
- (2)  $\tan \frac{\pi}{4} =$
- (3)  $\sin(15^\circ + 30^\circ \div 2) =$
- (4)  $\sin(-30^\circ) =$
- (5)  $\sqrt{3^2 + 4^2} =$
- (6)  $\sin^2 30^\circ =$
- (7)  $\log 5 + \ln 5 =$
- (8)  $e^{2+3} =$
- (9) 60進数の30度30分は10進数の何度?
- (10) 10進数の30.5度は60進数の何度何分何秒?

[解 答] (1)(3)(4)(6)では角度指定をDEGに、(2)では角度指定をRADにして計算。

(次ページ参照)

- |  |                  |
|--|------------------|
| (1) <b>SIN 30 + COS 60 * TAN 45</b> <b>ENTER</b>   | 1                |
| (2) <b>TAN (<math>\pi / 4</math>)</b> <b>ENTER</b> | 1                |
| (3) <b>SIN (15 + 30 / 2)</b> <b>ENTER</b>          | 0.5              |
| (4) <b>SIN - 30</b> <b>ENTER</b>                   | -0.5             |
| (5) <b>√ (3 ^ 2 + 4 ^ 2)</b> <b>ENTER</b>          | 5                |
| (6) <b>SIN 30 ^ 2</b> <b>ENTER</b>                 | 0.25             |
| (7) <b>LOG 5 + LN 5</b> <b>ENTER</b>               | 2.308407917      |
| (8) <b>EXP (2 + 3)</b> <b>ENTER</b>                | 148.4131591      |
| (9) <b>DEG 30.30</b> <b>ENTER</b>                  | 30.5<br>(30.5度)  |
| (10) <b>DMS 30.5</b> <b>ENTER</b>                  | 30.3<br>(30度30分) |

## 〔説 明〕

## ◎ 角度指定

三角関数、逆三角関数では、計算する角度の単位を正しく指定しておく必要があります。

角度単位	表示シンボル	指定のしかた	備 考
度	<b>DEG</b>	DEGREE <input type="button" value="ENTER"/>	直角を90で表わす単位〔°〕
ラディアン	<b>RAD</b>	RADIAN <input type="button" value="ENTER"/>	直角を $\frac{\pi}{2}$ で表わす単位〔rad〕
グ ラード	<b>GRAD</b>	GRAD <input type="button" value="ENTER"/>	直角を100で表わす単位〔g〕

$$(90^{\circ} = \frac{\pi}{2}(\text{rad}) = 100^{\text{g}})$$

- 指定された角度単位はシンボルとして表示部に表示されます。(DEG, RAD, GRAD)

## ◎ 関数機能

項 目	一般の表示	本機のキー操作	備 考
三 角 関 数	sin cos tan	SIN COS TAN	
逆 三 角 関 数	$\sin^{-1}$ $\cos^{-1}$ $\tan^{-1}$	ASN ACS ATN	
常 用 対 数	log	LOG	$\log_{10} x$ (10を底とする対数)
自 然 対 数	ln	LN	$\log_e x$ (eを底とする対数)
指 数 関 数	$e^x$	EXP	$e=2.718281828$
べき 乗		$\wedge$	$A \wedge B$ において $A^B$
開 平	$\sqrt{\quad}$	$\sqrt{\quad}$ あるいはSQR	
10進数(度)→60進数(度分秒)変換		DMS	角度変換
60進数(度分秒)→10進数(度)変換		DEG	
整 数 化 (インテジャー)		INT	INT(x)において、x以下でかつもっとも大きい整数を求めます。
絶 対 値 (アブソリュート)	X	ABS	ABS(x)において、xの絶対値を求めます。
符 号 関 数 (シグナム)		SGN	SGN(x)において、 $x > 0$ のとき1、 $x < 0$ のとき-1、 $x = 0$ のとき0が得られます。
円 周 率	$\pi$	$\pi$ あるいはPI	$\pi \approx 3.141592654$

## 〈関数の計算範囲〉

関数	計算範囲
$y \wedge x$ ( $y$ の $x$ 乗)	$-1 \times 10^{100} < x \log y < 100$ ( $y=0, x \leq 0$ のときは ERROR 39 $y=0, x > 0$ のときは 0 $y < 0, x \neq$ 整数のときは ERROR 39) 下記の例を参照
SIN $x$ COS $x$ TAN $x$	DEG: $ x  < 1 \times 10^{10}$ RAD: $ x  < \frac{\pi}{180} \times 10^{10}$ GRAD: $ x  < \frac{10}{90} \times 10^{10}$ ただし、TAN $x$ において次の場合は除く DEG: $ x  = 90(2n-1)$ RAD: $ x  = \frac{\pi}{2}(2n-1)$ GRAD: $ x  = 100(2n-1)$ ( $n$ は整数)
SIN $^{-1}x$ COS $^{-1}x$	$-1 \leq x \leq 1$
TAN $^{-1}x$	$ x  < 1 \times 10^{100}$
LN $x$ LOG $x$	$1 \times 10^{-99} \leq x < 1 \times 10^{100}$
EXP $x$	$-1 \times 10^{100} < x \leq 230.2585092$
$\sqrt{x}$	$0 \leq x < 1 \times 10^{100}$

上記以外の関数は、演算数  $x$  が次の範囲で計算可能です。

$$1 \times 10^{-99} \leq |x| < 1 \times 10^{100} \text{ および } 0$$

(例)  $0 \wedge 0$  [ENTER] → ERROR 39  
 $0 \wedge 5$  [ENTER] → 0  
 $(-4) \wedge 0.5$  [ENTER] → ERROR 39  
 $-4 \wedge 0.5$  [ENTER] → -2

- 関数の精度は原則として10桁目に±1（指数表示の場合は仮数部の最下位桁に±1）の誤差となります。

## ＝ご注意＝

PC-1500で三角関数の計算をおこなうとき

RADモードでは  $\frac{\pi \times n}{18 \times n}$  (18は1.8、180、1800…の)  
 GRADモードでは  $\frac{20 \times n}{18 \times n}$  (ような場合も同じ)

のような場合に、計算結果が0 (COSでは1) となることがわかりました。

この場合  $A = \pi/18$ 、 $B = 20/18$  のように、他の変数に代入したのち、SIN  $A$  のように計算をおこなえば正しく計算できます。

連続計算をおこなうときは、計算途中で上記のような計算となることも考えられますので、三角関数の計算をおこなう場合は、下記のように変数に代入したのち、計算をおこなってください。

〈例〉

通常の計算	変数に代入しての計算
SIN (X/Y)	A=X/Y SIN A

COS、TAN も同様です。



## 2-3) メモリー計算

## 〔例題〕

- (1) ① Aメモリーに3, Bメモリーに4を入れてください。  
 ②  $\sqrt{A^2+B^2}$ を計算してその結果をCメモリーに入れてください。  
 ③ Aメモリー, Bメモリー, Cメモリーの内容を順次確認してください。
- (2) ① 上のAメモリーの内容に1を, Bメモリーの内容に2を加えて, それぞれのメモリーに入れてください。  
 ②  $\sqrt{A^2+B^2}$ を計算して結果をCメモリーに入れてください。  
 ③ A, B, Cの各メモリーの内容を順次確認してください。
- (3) ① A\$メモリーに"PC-1500", B\$メモリーに"BASIC"という文字を入れてください。  
 ② A\$メモリー, B\$メモリーを呼び出して確認してください。

## 〔解答〕

- (1) ① A=3  } または A=3, B=4   
       B=4   
       ② C= $\sqrt{(A*A+B*B)}$    
       ③ A  3  
       B  4  
       C  5
- (2) ① A=A+1   
       B=B+2   
       ② C= $\sqrt{(A*A+B*B)}$    
       ③ A  4  
       B  6  
       C  7.211102551
- (3) ① A\$="PC-1500"   
       B\$="BASIC"   
       ② A\$  PC-1500  
       B\$  BASIC

## 〔説明〕

メモリーについての詳しい説明は42ページでの説明にゆずり, ここでは数値や文字を入れておくデータメモリーについて簡単に説明します。

データメモリーは次のように分類されます。(パーソナルコンピュータなど, 上級機ではデータメモリーのことを変数と呼ぶのが普通です。)

データメモリー(変数)  $\left\{ \begin{array}{l} \text{数値メモリー(数値変数)} \\ \text{文字メモリー(文字変数)} \end{array} \right.$

## ◎ 数値変数への数値の格納（代入）

A = 3

A = A + 1

のように、数値変数=式  と操作します。

変数の名前はA～Zだけでなく、AA、BZ、C5などのように2個の文字・数字を名前に使用することができます。この場合は、最初の文字が必ず英文字(大文字)でなければなりません。(ただし、PIやIFなどは計算機が関数などの命令と判断しますので使用できません。)

(例) AA = 5

ZC = 7 + 2

A1 = AA \* ZC

変数の名前に3個以上の文字・数字を使用してもかまいません。ただし、最初の2字以外は無視されます。

たとえば変数名にABCやKOTAEなどを使用することができますが、KOTAE1とKOTAE2では計算機には区別できません。

(最初の2字以外は無視されますので、計算機はKOという同じ変数と判断してしまいます)

- 数値変数には仮数部10桁、指数部-99～99までの数値を格納することができます。

## ◎ 文字変数への文字の格納

変数名の後に\$(ドルマーク)をつけることにより文字変数を表わし、

文字変数 = "文字"

または

文字変数 = 文字変数

と操作します。

(例) A\$ = "SHARP"

BZ\$ = "COMPUTER"

C3\$ = A\$

- 文字変数には最大16文字まで格納することができます。

### 3. プログラム(その1)

前項でいろいろなマニュアル計算に慣れていただきましたが、いろいろな計算の中でまったく同じような計算を何度もおこなうことがよくあります。

たとえば、円の面積を求める計算を例に取ってみましょう。円の面積 $S$ を求める式は

$$S = \pi r^2$$

でしたね。そして $r$ が半径です。

半径が3cmの円の面積は？

$\pi * 3 * 3$  **ENTER**

半径が4cmでは？

$\pi * 4 * 4$  **ENTER**

などと操作します。このように何度も使う計算の手順を計算機に覚えさせておけば、上例の $r$ の値のように、その都度変わる数を入力するだけで答が一度に求められるのです。

さらにいろいろな判断が必要になる複雑な計算も、その手順を計算機に覚えさせておけば、人間が順次判断するのと同様に計算機が判断して一度に答を求めてくれます。

計算の手順を計算機が理解できるように書き表わしたり、計算機に覚えさせることをプログラミングとかプログラムを組むなどと言います。

ここではプログラムの基本について説明します。

- プログラムを計算機に覚えさせるときはPROモードにしてください。

#### プログラミングの前に

電池交換をおこなった後などでは、計算機の状態などが一定せず、プログラムが書き込めないことがあります。したがって、はじめてプログラミングをおこなうときは最初に

**CL** **NEW** **0** **ENTER**

と押してください。

また、古いプログラムを消すときは

( **CL** ) **NEW** **ENTER**

と押してください。

- プログラム・データエリアはプログラムとデータ用として共用しています。(45ページ参照)  
プログラムとデータの合計が、プログラム・データエリアの容量を越えそうなときは、プログラムに **65279 END** と書き込んで、最終ラインを規定してください。

### 3-1) 簡単なプログラム (INPUT、LET、PRINT、END)

#### 〔例 題〕

半径を入力して円の面積を求めるプログラムを作ってください。

#### 〔解 答〕(プログラム1)

```
10 INPUT R
20 LET S=π*R*R      ←通常、LETは省略できます。
30 PRINT S
40 END
```

#### 〔説 明 1〕

BASICをまったく知らない人でもこのプログラムが計算機に何を伝えているのか想像がつくと思います。これは次のような指示をするプログラムなのです。

```
10 INPUT R      ←データを読み込み、Rに入れなさい。
20 LET S=π*R*R  ←πにRの値を2回掛けなさい。そして、その結果をSに入れ
                  なさい。
30 PRINT S      ←Sの値を表示しなさい。
40 END          ←これでプログラムは終了です。
```

さて、ここでプログラムの構成と各命令について少しくわしく説明します。プログラムの一番基本となるところですから十分理解するようにしてください。

		ライン
10	INPUT R	← (10ライン)
20	LET S=π*R*R	← (20ライン)
30	PRINT S	← (30ライン)
40	END	← (40ライン)

↑                      ステートメント  
ラインナンバー

#### ◎ ライン

BASICでは、プログラムはいくつかのラインから構成されています。そして1つのラインは、1つのラインナンバーと、いくつかの意味を持ったステートメント(命令など)からなります。(各ラインの終りは **ENTER** キーで指定します。)

↑                      ステートメント      **ENTER**  
ラインナンバー

( **ENTER** はラインの終りを意味し、計算機にプログラムを入れるときには必ず必要です。)



## ◎ ラインナンバー

ラインナンバーは1～65279の整数をつけます。

プログラムの実行は若いラインナンバーから順次実行されますので、計算の順に若いラインナンバーをつけます。ただし、あとからプログラムの途中でラインを追加したい場合も出てきますので、そういったことを配慮して10、20、30……のように10番おきにラインナンバーをつけるのが最も一般的です。

## ◎ ステートメント

1つの意味を持った処理式や命令をステートメントといいます。

1つのラインは1つ以上のステートメントからなり、2つ以上のステートメントになる場合はステートメントとステートメントの間に: (コロン)を入れて区別します。

ラインナンバー    ステートメント    ステートメント    :    ステートメント    ENTER

## ◎ INPUT命令 (インプット) (くわしくは、26 ページで説明します。)

プログラム実行中にマニュアル操作により数値や文字の入力をおこなうための命令で、与えられた数値や文字を入れる数値変数や文字変数をINPUTに続いて書きます。

INPUT A (変数Aに数値を入れなさい)

INPUT C\$ (変数C\$に文字を入れなさい)

なお、複数の数値や文字を連続して入れることもでき、次のように使います。

INPUT A, B

INPUT C\$, D\$

INPUT A, B, C\$

## ◎ LET命令 (レット)

代入文を意味し、LETに続いて代入式を書きます。

LET S =  $\pi$  \* R \* R

なお、LETは通常省略して使用できます。代入式は一般の数学とは少し異なり、右辺の数値あるいは数式を計算して左辺の変数に入れる(代入する)ことを意味します。したがって左辺は変数に限ります。

## ◎ PRINT命令 (プリント) (くわしくは、27 ページで説明します。)

PRINTに続いて変数を書いて、その変数の内容を表示させます。

PRINT A (Aの値が表示部の右端につめて表示されます。)

PRINT A\$ (A\$の内容が表示部の左端から表示されます。)

なお、変数の代わりに式や文字を書くこともできます。

PRINT 2 + 3

PRINT "KOTAE"

## ◎ END命令 (エンド)

BASICではプログラムの終りに“これでプログラムは終了です!” という意味でEND文を書きます。

## 3-2) プログラムの入れかたと編集 (NEW、LIST)

### ◎ プログラムの入れかた (書き込み)

先に作りましたプログラム(プログラム1)を計算機に書き込み、覚えさせましょう。

○まず、**MODE** キーを押して、表示部にPROシンボルを表示させてください。(PROモードにします。)

○次に、NEW **ENTER** と押してください。これで、先に記憶されている(かもしれない)プログラムやデータをすべて消します。そののち、65279 END **ENTER** と押してください。

○続いてプログラムを1ラインずつ順番に入れてください。

このとき注意していただきたいことは、1ラインを入れ終るごとに **ENTER** キーを押すことです。

たとえば 10 INPUT R **ENTER** と押します。

↑ **ENTER** は“このラインは終わりましたヨ”という意味で押します。

このように、1ライン入れ終るごとに **ENTER** を押すことを忘れないで、すべてのプログラムを入れてください。

PROシンボルを表示させます。  
**MODE**  
 NEW **ENTER**  
 10 INPUT R **ENTER**  
 20 LET S =  $\pi$  \* R \* R **ENTER**  
 30 PRINT S **ENTER**  
 40 END **ENTER**

プログラムを入れていただければ、おわかりと思いますが、各ラインを入れ、**ENTER** と押しますと、ラインナンバーの後に自動的に:(コロン)が入り、INPUTやLETなどの命令の後が自動的に1桁あけられます。

10: INPUT R  
 ↑ 自動的に1桁あきます。  
 ↑ 自動的に:が入ります。

このことにより、計算機が、最初の数字をラインナンバー、アルファベットを1つの命令(INPUTという命令)として受け取り(理解し)、覚えたことを示します。

### ◎ プログラムの編集

プログラムを作成する場合、一度で正しくは、なかなか書けないものです。キー操作のミスや勘違いでまちがえることもあります。このような場合の処理方法について次に説明します。

○表示されているライン内の訂正や追加、削除は13ページのマニュアル計算での説明と同じように **←** または **→** キーでカーソルを移動させ、**SHIFT** **DEL** や **SHIFT** **INS** を使っておこないます。

(例) 20 LET S= $\pi$ \*R\*R を

20 INPUT S= $\pi$ \*R\*R **ENTER** と押した場合の訂正

**▶** または **◀** を操作 20 INPUT S= $\pi$ \*R\*R

↑ここを点滅させます。

**SHIFT** **DEL** を操作 20 S= $\pi$ \*R\*R

↑ここが点滅します。

**SHIFT** **INS** を操作 20 **□□□** S= $\pi$ \*R\*R

↑必要な桁数分をあけます。

LET **ENTER** 20:LET S= $\pi$ \*R\*R

○ラインの追加は次のようにおこないます。

(例) プログラム1を計算機に入れるとき、次のように LET S= $\pi$ \*R\*R をぬかしたとき。

10 INPUT R

←この間に LET S= $\pi$ \*R\*R を追加したい。

20 PRINT S

30 END

10ラインと20ラインの間に追加する場合は、この間のラインナンバー(11~19のいずれか)を使ってプログラムを書きます。

たとえば 15LET S= $\pi$ \*R\*R **ENTER** とします。そうすればプログラムは次のように正しくなります。

10 INPUT R

15 LET S= $\pi$ \*R\*R

20 PRINT S

30 END

注) プログラムは若いラインナンバーから書く必要はなく、計算機が自動的に若いラインナンバーから順に並べかえてくれます。

注) ラインナンバーだけを変えたいときは **◀** キーでラインナンバーを点滅させて、そのナンバーを変更します。

(例) 30 END のラインナンバーを40に変えるとき

**◀** を操作 30 END

↑点滅させます。

4 **ENTER** 40:END

40ラインが書き込まれます。

30 **ENTER** > 30ラインを削除します。

○あるラインをすべて削除するときはラインナンバー **ENTER** と押します。

○プログラムの内容の確認は次の方法でおこなってください。


①  キー


このキーを1回押すごとに現在表示しているラインの次のラインを表示します。また、このキーを押し続けると、順次、次のラインを表示します。

②  キー

このキーを1回押すごとに現在表示しているラインの前のラインを表示します。また、このキーを押し続けると、順次前のラインを表示します。


③ LIST命令(リスト)

PROモードで LIST  と押せば、もっとも若いラインを表示します。


LIST \_\_\_\_\_   
ラインナンバー

と押せば、指示されたラインを表示します。(もし、指示されたナンバーのラインがない場合は、それよりも大きく、かつ一番近いラインが表示されます。)

●LIST命令はPROモードでのみ有効です。


注) 表示部には最大26桁表示することができますが、1ラインに長いプログラムを書きますと、その一部は表示できなくなります。このような隠れた部分を見るときは  キーを押し続けてください。そうすればカーソルが右端まで行き、そののち表示が左に送られて隠れている部分が出てきます。

### 3-3) プログラムの実行 (RUN)

○プログラムの実行はRUNモードでおこないますので、 キーを押して表示部にRUNシンボルを表示させてください。

○続いて次の操作により実行を開始します。





RUN  (最も若いラインナンバーより実行)

RUN \_\_\_\_\_   
ラインナンバー (指定したラインナンバーより実行)

(例) プログラム1 (21ページ) を実行しましょう。まずプログラムを計算機に入れてから次の計算をしてください。

①半径が5cmの場合

②半径が10cmの場合

 >  
① RUN  ?  
5 5\_  
  
 >

RUNシンボルを  
表示させます。  
↓  
RUN

78.53981634



```

② RUN [ENTER]      ?
          10         10_
          [ENTER]
          [ENTER]      >
                                     314.1592654

```

○上例のようにINPUTでは?を表示して実行をいったん停止しますので、データを入力して [ENTER] キーで実行を再開させます。

○INPUT命令によりプログラム実行が停止しているとき、データを入力せずに [ENTER] キーを押した場合は、そのINPUT命令が書かれているラインのINPUT命令以降の命令を無視(スキップ)し、次のラインへ実行に移ります。

○INPUT命令において、文字変数に文字を入れるときは“ ”で囲む必要はありません。また、文字変数の内容を消去するときは、INPUT命令によりデータの入力が求められているときに、” だけを入力して [ENTER] キーを押してください。(このとき表示部上側のシンボル以外は、なにも表示されません。)

(注) RUN命令でプログラムを実行したとき、それまで確保されていた2文字の変数は消されます。(41ページを参照)

## ＝エラー表示と処理方法＝

プログラムを実行しているときにエラーが発生しますと、

```
ERROR 39 IN 30
```

のように、エラー番号と、エラーが発生したラインナンバーを表示します。

このときは、次のようにしてエラー処理をおこなってください。

- ① [CL] キーでエラーを解除してください。  
 (エラーが表示されているとき、あるいは [CL] でエラーを解除したとき、 [↑] キーを押せば、押している間だけエラーが発生したプログラムラインを表示します。)
- ② [MODE] キーを押してPROモードにしてください。
- ③ [↓] キーあるいは [↑] キーを押してください。エラーが発生したプログラムラインが表示され、エラーが発生した位置がカーソルで示されています。
- ③ エラーが発生した原因を探して、13ページおよび、24 ページに記載の方法で訂正などをおこなってください。

●エラーの内容については167～169ページを参照してください。

## 3-4) いろいろな入力形 (INPUT)

3-3項でプログラムの実行をおこないましたが、入力・出力が味気ない感じがしたでしょう。この項と次の項では入力や出力に変化を持たせる入力形・出力形の基本を説明します。

### 〔例 題〕

半径入力時に HANKEI R= と表示させるような円の面積を求めるプログラムを作成してください。

## 〔解答例〕

```

10 INPUT "HANKI R=", R
20 S=π*R*R
30 PRINT S
40 END

```

↑ 本書ではスペースを意味しています。計算機に入るときは **SPACE** キーを操作してください。

○計算機に書き込んで実際に実行し、表示を確認してください。また、10ラインを次のように変えて実行してみてください。

```

10 INPUT "HANKI R=";R

```

↑ この部分を,(コンマ)から;(セミコロン)に変えます。

## 〔説 明〕

◎ INPUT時にメッセージを表示させる方法です。

① INPUT "文字", 変数

(例) INPUT "A=", A  
INPUT "A=", A, "B=", B

② INPUT "文字"; 変数

(例) INPUT "A="; A  
INPUT "A="; A, "B="; B

①の形ではINPUT状態で数値を入力すればメッセージが消えますが、②の形ではメッセージは消えずに、入力したデータがメッセージに続いて表示されます。

(注) INPUT命令実行時、データの入力誤りなどでエラーになった場合は

**SHIFT** **CL** と押してエラーを解除し、改めて最初からプログラムを実行してください。  
なお、**CL** キーでエラーを解除した場合、エラー表示の一部が消えずに残ることがありますが、そのままデータの入力などをおこない、計算を続けることができます。  
(実行には影響ありません。)

(注) 51ページで説明します G\_CURSOR命令で151以上が指定されているとき、続いて INPUT命令がある場合は、入力された内容が表示できませんのでご注意ください。

## 3-5) いろいろな出力形

## 〔例 題〕

- ①3-4項での例題において、円の面積を表示するとき、左半分に半径、右半分に面積を表示させるプログラムにしてください。
- ②また、円の面積を表示するときに S= と表示させ、それに続いて面積を表示させるプログラムにしてください。

## 〔解答例〕

```

① 10 INPUT "HANKI R=", R
20 S=π*R*R
30 PRINT R, S
40 END

```

```

② 10 INPUT "HANKEI,R=",R
    20 S=π*R*R
    30 PRINT "S=";S
    40 END

```

○プログラムを実際に行ってみてください。また、30ラインを次のように変えて実行し、その違いをみてください。

- ① 30 PRINT R;S \_\_\_\_\_, (コンマ)を;(セミコロン)に変えます。  
 ② 30 PRINT "S=";S \_\_\_\_\_; を,に変えます。

# [説明]

◎ PRINTは基本的に次の使いかたができます。

## ① PRINT式

この形では式の値を表示部の右端を使って16桁以内で表示します。

注) 式とは、数値や数値変数およびこれらを含んだ式をさします。

(例) 3, A, 3\*4, 3\*A+5など

(例) PRINT S      Sの内容を右端16桁以内に表示

## ② PRINT文字変数

PRINT "文字"

この形では文字あるいは文字変数の内容を表示部の左端から表示します。

(例) PRINT A\$      A\$の内容を左端より表示

PRINT "MENSEKI,S="      文字を左端より表示

③ PRINT 

式
"文字"
文字変数

 , 

式
"文字"
文字変数

26桁の表示部を左右13桁に分けて、.の左右の内容を表示させます。このときも、13桁の範囲内で数値は右ツメ、文字は左ツメにします。なお、表示内容が13桁を超える場合は、頭から13桁のみを表示します。

(例) PRINT A, S      Aの値を左13桁内に、Sの値を右13桁内に右ツメで表示。

PRINT "S=",S      S=を左13桁内に左ツメで、Sの値を右13桁内に右ツメで表示。

④ PRINT 

式
"文 字"
文字変数

 ; 

数値変数
"文 字"
文字変数

 ; ... ; 

数値変数
"文 字"
文字変数

この形では文字や数値を表示部の左端よりツメで26桁の範囲内で表示します。

(例) PRINT "S="; S      左端にS=を表示し、続いてSの内容を表示します。

## 4. プログラム(その2)

プログラム(その1)では、プログラムの一番基本となる入力、計算、出力の3点について説明しましたが、ここではそれに加えてジャンプや判断、くり返し計算などを説明します。プログラムの大部分はこれらジャンプ、判断、くり返しなどでなっています。いわばプログラムの本当の基本となるものです。十分理解するようにしてください。

### 4-1) ジャンプと判断 (GOTO, IF...THEN)

#### (1) ジャンプ

##### 〔例題〕

劇場に入場する人数を入力して、その料金を求めるプログラムを作ってください。

ただし、料金は1人 1,400円として、表示は左側に人数、右側に料金を表示させてください。

```
NINZUU      8      11200 EN
```

##### 〔解答例〕(プログラム2)

```
10: INPUT "NINZUU="
    ";A
20: B=1400*A
30: PRINT "NINZUU_
    _";A;" _";B;"
    _EN"
40: GOTO 10
```

(注) このプログラムは10ラインから40ラインまでをくり返し実行するようになっていますので、プログラムの実行が終了しません。このようなプログラムにおいて実行を終了させるときは **BREAK** **ON** **CL** と押してください。

##### 〔説明〕

解答例のプログラムは、40ラインを除けばこれまで説明しましたプログラムと同じ形です。

ここでは40ラインに書かれていますGOTO命令について説明します。

#### ◎ GOTO命令(ゴートゥー)

プログラムの実行は若いラインナンバーから順次実行されますが、このGOTO命令を実行すると、GOTOに続いて書かれているラインへジャンプして、ジャンプしたラインから実行します。

(例)	GOTO 40	ラインナンバー40へジャンプ
	GOTO 5*8	ラインナンバー40へジャンプ
	GOTO A	Aの内容のラインへジャンプ(A=40ならば40ラインへジャンプ)
	GOTO "A"	"A" ラベルのついているラインへジャンプ(次ページ参照)



<sup>ラベル</sup>  
 (例) 10: "A": INPUT "NINZUU="; A  
 20: B=1400\*A  
 30: PRINT "NINZUU\_"; A; "\_\_\_"; B; "  
 \_EN"  
 40: GOTO "A"

40ラインから10ラインへジャンプ  
 このようなジャンプをラベルジャンプといいます。

ラベル：ジャンプ先につけた名前のことです。GOTO命令などでラベルジャンプが指定されていますと、実行の際、そのラベルを探してそのラインにジャンプします。ラベルジャンプの場合、プログラムの訂正などでラインナンバーをつけかえても、プログラムは変更する必要がないので便利です。なお、同じラベルが2個以上書かれているときは、ラインナンバーの小さいほうへジャンプします。

## (2) 判断

### 〔例題〕

(1)項の例題において、人数が10人未満のときは1人あたりの料金を1,400円に、10人以上のときは団体料金として1人あたりの料金を1,200円にして料金を求めるプログラムを作ってください。

### 〔解答例1〕

```

10: INPUT "NINZUU="
   ";A
20: IF A>=10 THEN 40      ← 人数が10人以上かどうか判断
   0                      (10人以上のときは40ラインへジャンプ)
30: B=1400*A: GOTO 50
   50
40: B=1200*A
50: PRINT "NINZUU_"; A; "___"; B; "
   _EN"
60: GOTO 10
  
```

### 〔解答例2〕

```

10: INPUT "NINZUU="
   ";A
20: IF A>=10 LET B=
   1200*A: GOTO 40
30: B=1400*A
40: PRINT "NINZUU_"; A; "___"; B; "
   _EN"
50: GOTO 10
  
```

### 〔説明〕

#### ◎ IF命令(イフ)

IF命令はいろいろな判断をおこなう命令です。たとえば

```
IF A >= 10
```

の例では、Aの値が10以上(IF文が成立)ならばIF文の後に続く内容を実行し、10未満、(IF文が不成立)ならば次のラインを実行します。

IF文に続く、A >= 10などの式を条件式といいます。条件式は通常次のような形で書きます。

IF ○○=××	等しいかどうか判断 (○○は××に等しいか?)
IF ○○>××	大きいかどうか判断 (○○は××より大きいか?)
IF ○○>=××	以上かどうか判断 (○○は××以上か?)
IF ○○<××	小さいかどうか判断 (○○は××より小さいか?)
IF ○○<=××	以下かどうか判断 (○○は××以下か?)
IF ○○<>××	等しくないかどうか判断 (○○と××は等しくないか?)

(注) ○○および××は式、変数を表わします。(5\*4、A、8など)

IF文の後には通常THEN60のように使い、指定したライン(60ライン)へのジャンプを指示します。(THEN:ゼン)なお、解答例2のようにIF文の後にステートメントを書いたほうがわかりやすい場合があります。(IF文の後に続いて、代入文を書く場合は必ずLETが必要です。)

条件式を通常の式にすることも可能です。たとえば

IF A>B+4 THEN 60

とした場合、Aの値がB+4の式の値よりも大きいならば60ラインを実行し、Aの値がB+4の式の値よりも大きくないならば、次のラインを実行します。

#### ＝参 考＝

論理演算子(=、>、<、>=、<=、<>)を用いて構成された式(論理式)は、それぞれ次の値をとります。

x、yは数値を示します。

※ =	x=yのとき:1 x≠yのとき:0	>=	x≥yのとき:1 x<yのとき:0
>	x>yのとき:1 x≤yのとき:0	<=	x≤yのとき:1 x>yのとき:0
<	x<yのとき:1 x≥yのとき:0	<>	x≠yのとき:1 (<>は≠と同意味) x=yのとき:0

※ =を用いる論理式を例えばA=数値、B=式 のような形にしますと、計算機は論理式と判断せず、変数への入力(代入文)と判断しますので、このような場合は数値=A、式=B のような形にしてください。ただし、IF文における条件式の場合はこの限りではありません。

## =ご注意=

本機は仮数部が最大10桁までの計算をおこなうことができますが、内部では計算の精度を上げるために仮数部12桁まで計算をおこない、11桁目を四捨五入処理したうえで計算結果として表示します。例えば  $5 \div 9$  および  $5 \div 9 * 9$  の計算は内部で次のように計算されます。

```

5 / 9  →  5.5555555555E-01
           ↑11桁目を四捨五入
表示   →  5.55555556E-01

5 / 9 * 9 →  4.9999999999E-00
           ↑11桁目を四捨五入
表示     →           5

```

このように内部では、あくまで仮数部12桁の数値を用いて計算がおこなわれています。したがって、計算を連続しておこなった場合と、区切って計算した場合とでは、その結果に差が生じることがあります。

(例1)  $3^2 - 9 =$ 

〈連続した計算〉

3 **SHIFT** **^** 2 **=** 9 **ENTER** → -9 E-11

〈区切った計算〉

3 **SHIFT** **^** 2 **ENTER** → 9  
**=** 9 **ENTER** → 0

また、IF文においても、条件式が連続的な計算では、この差のためにプログラムが目的通りに働かない場合も生じますのでご注意ください。

## (例2) 10 INPUT A

20 IF  $A^2 \geq 9$  THEN 50

⋮

Aが3のとき、 $3^2$ が計算機内部では8.9999999999E-00 となるためIF文が成立しません。

このようなときは、次のように変数を用いて〈区切った計算〉になるようにプログラムしてください。

10 INPUT A

15 B =  $A^2$

20 IF B  $\geq$  9 THEN 50

⋮

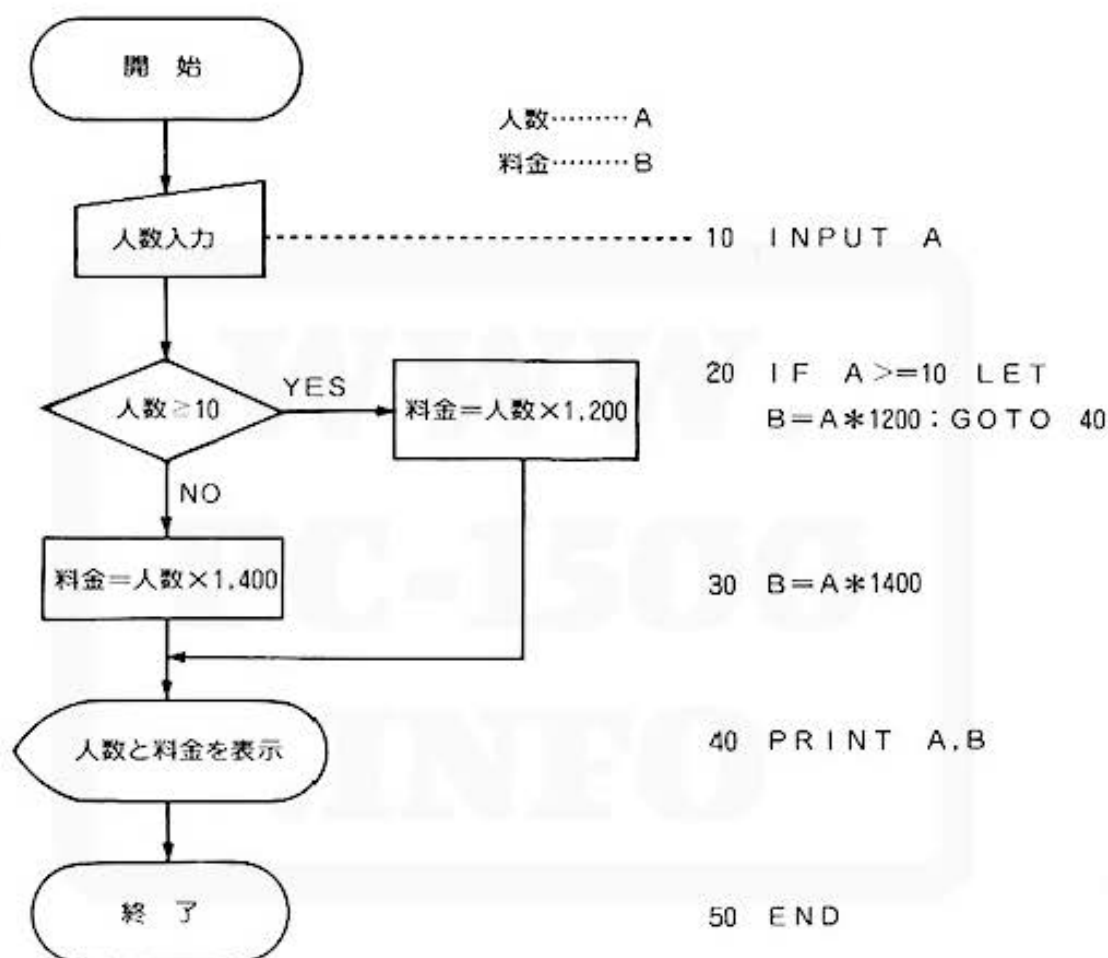
(べき乗計算は、 $\log x$ と $10^x$ 計算を用いて計算されるため、  
 計算機内との差が出やすくなります。  
 $A^B \rightarrow 10^{B \log A}$ により計算)

}  $A^2$ の結果を変数に代入し、その変数を用いて条件式を構成します。

## ＝プログラムの考えかた＝

問題を見ていきなりプログラムを作ることは容易なことではありません。手順として計算順序の概略を図式化し、解析するのが普通であり、これを流れ図とかフローチャートといいます。フローチャートができあがればプログラムはほぼできたといってもよいほどです。プログラム上達のためには、まずフローチャートに慣れるのが早道ではないでしょうか。

フローチャートの次に各データをどの変数に割当てていくが決めます。そして次にフローチャートの各部分をBASIC命令におきかえていき、プログラムを作成します。



解答例ではいろいろな文字表示など、使いやすく、見やすくするためにつけていますが、これらは一種のアクセサリーと考えてもよいでしょう。まず上のようなフローチャートを作り、上のようなプログラムを作ることが第一です。慣れるにしたがって、いろいろなアクセサリーに工夫をこらせばよいでしょう。

## 〔例題〕

金額と日数を入力して利息を求めるプログラムを作ってください。

ただし、日数が30日以下は日歩2銭、30日を越える分については日歩4銭とします。

## 〔解答例〕

## ● 解析

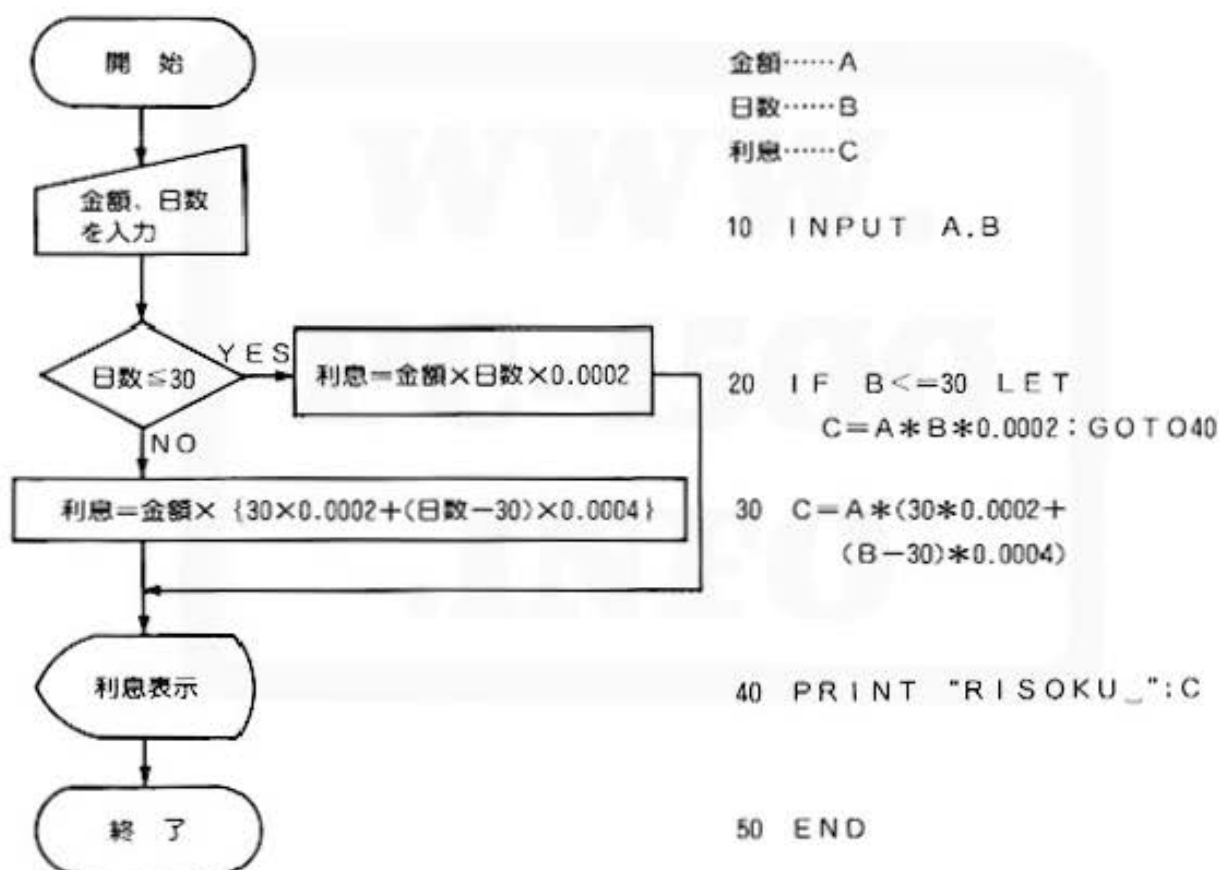
日数が30日以下のとき

$$\text{利息} = \text{金額} \times \text{日数} \times \frac{0.02}{100}$$

日数が30日を越えたとき

$$\text{利息} = \text{金額} \times \left\{ 30 \times \frac{0.02}{100} + (\text{日数} - 30) \times \frac{0.04}{100} \right\}$$

## ● フローチャートとプログラム





## 4-2) くり返し計算 (FOR...TO...STEP, NEXT)

### 〔例題〕

$\sin 0^\circ$  から  $\sin 90^\circ$  まで5度きざみで計算し、その結果を表示させるプログラムを作ってください。

### 〔解答例〕

```
10:FOR A=0 TO 90
    STEP 5
20:B=SIN A
30:PRINT "SIN";A;
    "=";B
40:NEXT A
50:END
```

FOR-NEXT ループ

(RUN **ENTER** で実行を開始したあとは、**ENTER** キーでつぎつぎに実行させます。)

### 〔説明〕

#### ◎ FOR.....NEXT文(フォー、ネクスト)

FOR NEXT文の一般形は次のようになります。

```
FOR 数値変数=初期値 TO最終値 STEPきざみ
{
NEXT 数値変数
```

数値変数が初期値から始まって、指定されたきざみずつ増加していき、最終値以上になるまでFORとNEXTの間の計算をくり返し実行します。(このくり返し部分をFOR-NEXTループと呼びます。)

- FORとNEXTは必ず対にして使い、FORの後の数値変数とNEXTの後の数値変数は同一の変数でなければなりません。
- きざみが1のときはSTEP 1は省略できます。
- 初期値、最終値、きざみを負の値にすることもできます。

(例) 10 FOR A=0 TO -90 STEP -5

#### ◎ FOR-NEXTループの注意事項

FOR-NEXTループを使う場合に注意すべきことがいくつかありますので次に説明します。

- ① 最終値ときざみ(ステップ数)は次の範囲内でなければなりません。

-32768~32767 (ただし、きざみ≠0)

きざみの値に小数部が含まれる場合、小数部は無視されます。

- ② FOR-NEXTループの中に、別のFOR-NEXTループを入れることができます。ただし、中に入るFOR-NEXTループは外のFOR-NEXTループ内に完全に入っていないとなりません。

(例1)

```

10  FOR A=1 TO 5
30  FOR B=0 TO 5
70  NEXT B
100 NEXT A

```

ループ①  
ループ②

ループ①がループ②の中に完全に入っていますので、この使い方は正しい。

(例2)

```

10  FOR C=1 TO 10
40  FOR D=1 TO 12
90  NEXT C
110 NEXT D

```

ループ①  
ループ②

ループ①とループ②が交差していますのでこの使い方は誤り (ERROR 2)

なお、FOR-NEXTループは3段、4段……と重ねて使う(深みをもたせる)ことができますが、いずれも、中に入るループは外のループ内に完全に入っていないとなりません。

(例)

```

10  FOR A=1 TO 5
40  FOR B=0 TO 6
100 FOR C=1 TO 12
120 FOR D=1 TO 10
150 NEXT D
190 NEXT C
210 NEXT B
300 NEXT A

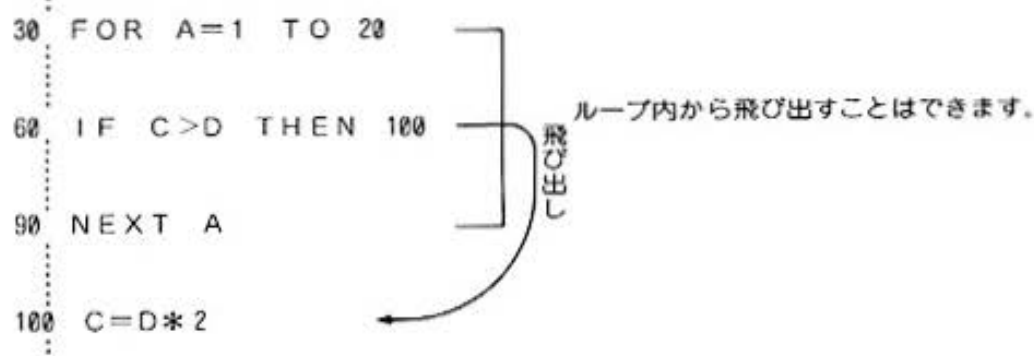
```

ループ①  
ループ②  
ループ③  
ループ④

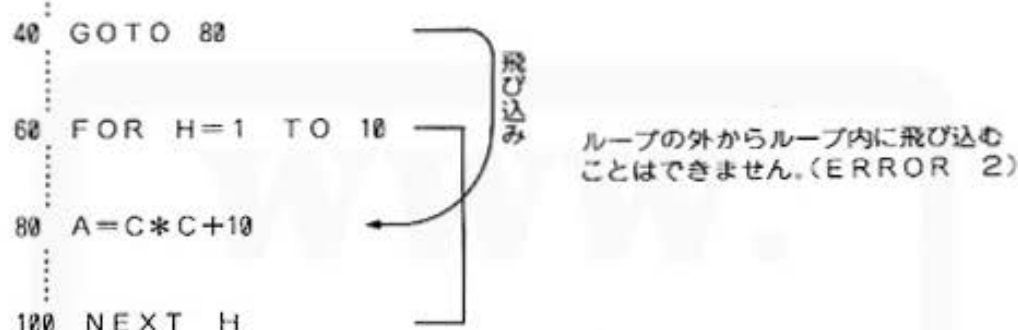
FOR-NEXTループは重ねて使うことができます。(段数については41ページを参照)

- ③ FOR-NEXTループ内から外に飛び出すことはできますが、外からFOR-NEXTループ内に飛び込むことはできません。

(例1)

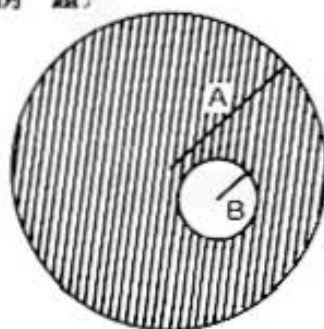


(例2)



### 4-3) サブルーチン (GOSUB, RETURN)

(例題)

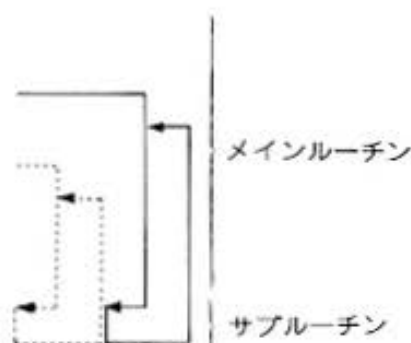


図のような2円があるとき、2円の半径を入力して斜線部の面積を求めるプログラムを作ってください。

(解答例)

```

10: INPUT "A="; A, "
    B="; B
20: R=A: GOSUB 70
30: C=S: R=B: GOSUB
    70
40: D=S
50: PRINT C-D
60: END
70: S=PI*R*R
80: RETURN
    
```



〈使用メモリー〉

A 大円の半径  
B 小円の半径  
C 大円の面積  
D 小円の面積

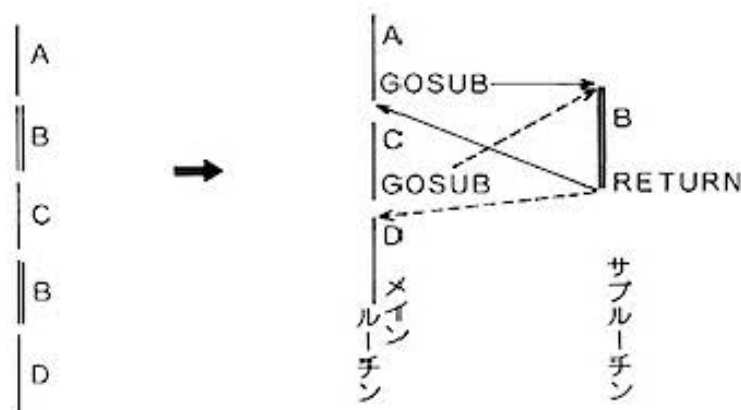
サブルーチン

R 円の半径  
S 円の面積

## 〔説 明〕

○何度も同じような処理をおこなうときに、その処理をサブルーチン化すると便利になります。

この例題はほんの一例ですが、共通処理部をたびたび使ったり、共通処理部が長い場合には全体のプログラムを短縮でき、またプログラムが見やすく、それだけ誤りも少なくなります。



上の左図のように共通部(B)があるとき、その部分をサブルーチンにすると右のようになります。サブルーチンへのジャンプはサブルーチンのおかれているラインナンバー(またはラベル)をGOSUBの後に続いて書いて指示します。またサブルーチンはRETURN命令でメインルーチン(GOSUBのあった元のプログラムをサブルーチンに対しメインルーチンと呼びます。)のGOSUBのあった次の命令を引続き実行します。

- サブルーチンの中にサブルーチンを入れることもできます。



- GOSUB命令の一般形は次のようになります。

GOSUB式

あるいは

GOSUB“文字”

GOSUB命令の指定の形はGOTO命令と同様です。

## 4-4) 定義付けプログラム

複数のプログラムが書き込まれているときに、2番目以降のプログラムを実行するときや、プログラムの途中から実行するときには、通常次の方法でスタートさせます。

- ① RUN    ラインナンバー    **ENTER**
- ② GOTO   ラインナンバー    **ENTER**

この方法により、指定したラインからプログラムの実行が開始されます。この方法では操作に手間がかかり、また、ラインナンバーを常に知っていなければならない、なかなかわずらわしいものです。このようなとき、プログラムラインの始めに "A" や "B" のようにラベルを書いておくことにより、**DEF** **A** や **DEF** **B** と押すだけでプログラムの実行を開始させる方法があります。

### 〔例 題〕

次の3つの計算をおこなうプログラムを作ってください。このとき、①のプログラムは **A** キーに、②のプログラムは **S** キーに、③のプログラムは **D** キーにそれぞれ定義付けをおこなって、**DEF** **A** 、 **DEF** **S** 、 **DEF** **D** と押すことで、それぞれのプログラムを実行できるようにしてください。

- |  |          |
|--|----------|
| ① $c = \sqrt{a^2 + b^2}$                   | ピタゴラスの定理 |
| ② $V = \frac{4}{3} \pi r^3$                | 球の体積     |
| ③ $c = \sqrt{a^2 + b^2 - 2ab \cos \theta}$ | 余弦定理     |

### 〔解答例〕

```

      ラベルA
10: "A": INPUT "A="
   ; A, "B="; B
20: C=√(A*A+B*B)
30: PRINT "C="; C
40: END
      ①

      ラベルS
50: "S": INPUT "R="
   ; R
60: U=4/3*π*R^3
70: PRINT "U="; U
80: END
      ②

      ラベルD
90: "D": INPUT "A="
   ; A, "B="; B, "Z="
   ; Z
100: C=√(A*A+B*B-2*
      A*B*COS Z)
110: PRINT "C="; C
120: END
      ③
  
```



## 〔説 明〕

## ◎ ラベル

ラベルは飛び越しの行先につけた名前(名札)のことです。前例のように **A** , **S** , **D** などのキーに対応する文字・記号を“ ”で囲ってラインナンバーのすぐ後に書き込んでおけば、**DEF A** , **DEF S** , ……と押したときに、そのキーに対応するラベルを探して、そのラベルの書かれているラインからプログラムが実行されます。

## ◎ 定義付けキー

定義付けのできるキーは次の18キーです。

**A** , **S** , **D** , **F** , **G** , **H** , **J** , **K** , **L** ,  
**Z** , **X** , **C** , **V** , **B** , **N** , **M** , および **=** , **SPACE**

注) 定義付けキーによりプログラムの実行開始を指定したとき、そのラベルをつけた(定義付けされた)プログラムがない場合はエラー(ERROR 11)になります。このときは **ON** キーあるいは **CL** キーでエラーを解除してください。( **▶** , **◀** キーでも解除できます。)

## ＝よく使う命令の簡単な入れかた＝

これまで、INPUTやPRINTなどの命令はキーを1つずつ押して、すべての文字を入れていましたが、次のようなことを知っていれば、操作を簡略することができます。

- ① アルファベットのキーのうち、次のキーにはそれぞれ次の命令が割り当てられています。

<b>Q</b>	<b>W</b>	<b>E</b>	<b>R</b>	<b>T</b>	<b>Y</b>	<b>U</b>	<b>I</b>	<b>O</b>	<b>P</b>
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
INPUT	PRINT	USING	GOTO	GOSUB	RETURN	CSAVE	CLOAD	MERGE	LIST
						※	※	※	

これらのキーを **DEF** キーに続いて押せば、割り当てられている命令が入力されます。

- ※ CSAVE、CLOAD、MERGEの各命令はカセットテープレコーダー用の命令であり別売のカラーグラフィックプリンタ(カセットインターフェイス内蔵)を取りつけたときにのみ使用することができます。このため、カラーグラフィックプリンタを取りつけていないときは、これらの命令は入力できず、替りに～シンボルが入力されます。
- ② 各命令には省略した形(省略形)で入力できるものがあります。たとえば、PRINTはP、GOSUBはGOS、と入力すれば計算機はそれぞれの命令として理解することができます。これらの省略形は171 ページの関数・命令一覧に記載していますので参照してください。

## ＝スタックエリアについて＝

(忙しい人は、この項は飛ばして、次へ進んでください。)

本機はプログラムを実行する際、FOR-NEXTやGOSUB-RETURN命令を一時記憶しておくための、あるいは計算の優先順位やカッコにより保留される、計算命令やデータを一時記憶しておくためのスタックエリアを持っています。

プログラム実行中あるいは計算実行中に、すぐに実行してしまうことのできない命令や計算をこのスタックエリアに一時記憶させておいて、実行できるようになったときに呼び出して実行します。このスタックエリアの容量は196バイトあります。

次にFOR-NEXTやGOSUB-RETURNなどのスタックエリアでの使用バイト数を表示します。

命令データ	使用バイト数
FOR-NEXT	12
GOSUB-RETURN	6
データ	8
計算命令(+、-、*、/ など)	2

FOR-NEXTやGOSUB-RETURNを何段も深みを持たせた場合や何重にもカッコを使用した計算をおこなった場合は使用バイト数が196バイトを越えてエラーになることがあります (ERROR 14, 15)

## ＝プログラムの実行開始方法によるスタート状態の違いについて＝

(忙しい人はこの項を飛ばして、次へ進んでください。)

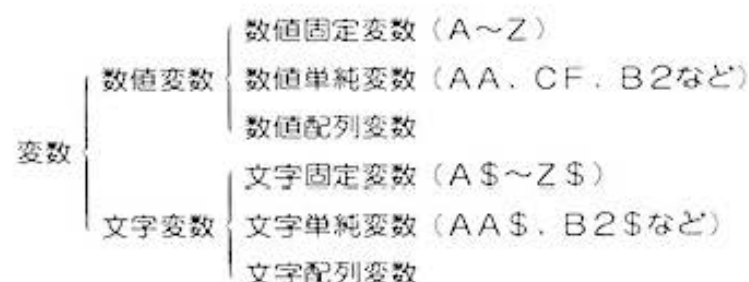
プログラムの実行開始方法には、RUN命令によるもの、GOTO命令によるもの、**DEF**  
**A**、などの定義付けキーによるものがありますが、これらの開始方法により、変数のクリアや状態の解除などに違いがありますので次に示します。

RUN	GOTO	定義付けキー
<b>〈実行方法〉</b> RUN <b>ENTER</b> RUN式 <b>ENTER</b>	<b>〈実行方法〉</b> GOTO <b>ENTER</b> GOTO式 <b>ENTER</b>	<b>〈実行方法〉</b> <b>DEF</b> に続く A, S, D, F, G, H, .....
○表示をクリア ○カーソルクリア ○ウェイト指定解除 ○表示フォーマット指定解除 ○配列変数 および2文字変数クリア ○FOR-NEXT、 GOSUBスタックのクリア ○ON ERROR解除 ○READ文に対するDATA の初期化	○表示をクリア ○FOR-NEXT、 GOSUBスタックのクリア	○FOR-NEXT、 GOSUBスタックのクリア

## 5. 変数

変数については2-3)でふれましたが、ここでは変数の構成などについて、さらに詳しく説明します。

### 5-1) 変数の構成



数値配列変数、文字配列変数については、のちほど5-2)で説明します。

#### ◎ 数値変数

数値変数は仮数部10桁、指数部-99~99で表わすことのできる数値を扱います。

##### ○ 数値固定変数

A~Zの変数

##### ○ 数値単純変数

AA, AB, F2, H3など、2文字の変数

変数の名前には、英大文字と英大文字、あるいは英大文字と数字の組み合わせ(1文字目が必ず英大文字)を用いることができます。(3文字以上の名前をつけた場合は前2文字が有効となります。)

#### ◎ 文字変数

次の文字変数は最大16字までの文字・記号を扱います。

##### ○ 文字固定変数

A\$~Z\$の変数

##### ○ 文字単純変数

AA\$, AB\$, F2\$, H3\$など、変数名(2文字)の後に\$記号をつけた変数

### 5-2) 配列変数 (DIM)

ある同数のデータをたくさん扱う場合がありますが、このようなときに配列変数を用いますと大変便利です。

#### 〔例題〕

INPUT文を使用して、任意の数値を10個読み込ませ、1番大きな数値を探し出すプログラムを作ってください。

```

10: DIM A(9)
20: FOR B=0 TO 9
30: INPUT "DATA=";
   A(B)
40: NEXT B
50: M=A(0)
60: FOR B=1 TO 9
70: IF M>=A(B) THEN
   90
80: M=A(B)
90: NEXT B
100: PRINT "SAIDAIL_
   WA_";M
110: END

```

FOR-NEXTループ  
データの読み込み

FOR-NEXTループ  
データの比較

## 〔説 明〕

- ◎ 配列変数を使うときは、事前にその変数として使う場所をメモリーの中に確保する必要があります。配列変数の場所を確保するためにはDIM(ディメンジョン)命令を使って、その変数名と大きさを定義します。たとえばDIM A(9)は配列変数A( )のためにA(0)からA(9)までの10個の変数がメモリー内に確保されることを意味します。(下図参照)

A(0)	A(1)	A(2)	A(3)	A(4)	A(5)
A(6)	A(7)	A(8)	A(9)		

- ◎ A(0)、A(1)…のように書かれている変数の、カッコの中の数字を添字といいます。本機では配列変数に添字が1個のものと2個のものを使用することができ、それぞれを一次元配列、二次元配列といいます。

一次元配列

DIM A(5) A(0)、A(1)……A(5)の6個の変数を確保

二次元配列

DIM A(3, 4) A(0, 0)、A(0, 1)、A(0, 2)……A(3, 4)の20個の変数を確保(下図参照)

A(0,0)	A(0,1)	A(0,2)	A(0,3)	A(0,4)
A(1,0)	A(1,1)	A(1,2)	A(1,3)	A(1,4)
A(2,0)	A(2,1)	A(2,2)	A(2,3)	A(2,4)
A(3,0)	A(3,1)	A(3,2)	A(3,3)	A(3,4)

## ◎ 文字配列

配列変数にも文字変数があり、固定変数や単純変数と同様に\$記号をつけて指定します。

(例) `DIM A$(9)`      `A$(0), A$(1).....A$(9)`の10個の変数を確保  
       `DIM A$(5, 4)`      `A$(0, 0), A$(0, 1).....A$(5, 4)`の30個の  
       変数を確保。

## ◎ 文字配列の拡張

文字固定変数や文字単純変数は、1個の変数に最大16文字まで記憶できる固定長の変数でしたが、配列変数では最大80文字以内で、任意に変数の長さを指定することができます。

(例) `DIM A$(9)*30`      `A$(0), A$(1),...A$(9)`の変数にはそれぞれ最大30文字まで格納することができます。  
       `DIM A$(5, 4)*6`      `A$(0, 0), A$(0, 1).....A$(5, 4)`  
       の変数にはそれぞれ最大6文字まで格納することができます。

上例のように、`DIM`文に \*式 をつけて変数の長さを指定します。指定できる文字数は1～80文字で、指定しない(\*式 がない)ときは自動的に16文字に設定されます。

- ◎ 配列変数は`A(B)`、`B$(C)`、`A1(C, D)`のように、添字を変数や式で指定することもできます。ただし、その値は0～255以内でなければなりません。また、基本のメモリー構成の場合、数値変数は0～229以内でなければなりません。

- ◎ 配列変数をたくさん定義する場合は次のような形で一度に定義できます。

`DIM A(20), B(5, 7), C$(10, 3)*6`

注) 同一プログラム内で、同じ変数名の配列を再定義することはできません。

`DIM A(20)`と`DIM A(3, 4)`は同じ`A( )`という変数名になります。

## ＝参 考＝

固定変数`A～Z`、`A$～Z$`も添字つきの変数として指定することができます。

`@(1)～@(26).....`数値変数`A～Z`を添字つきの変数として指定

`@$(1)～@$(26).....`文字変数`A$～Z$`を添字つきの変数として指定

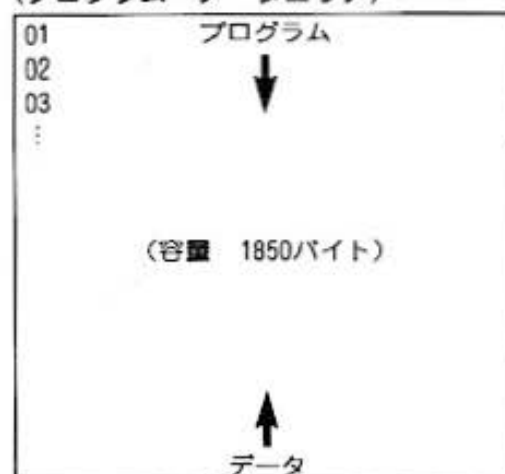
## 5-3) メモリーの構成と変数

変数については2-3)で少しふれましたが、ここでは本機の持つメモリーの構成と、変数についてさらに詳しく説明します。

### ◎ メモリーの構成



### 〈プログラム・データエリア〉



プログラムがこちら側から書き込まれていきます。

AAやA1\$のような2文字の変数、A(1)やB(5, 8)のような配列変数はこちら側から確保されます。

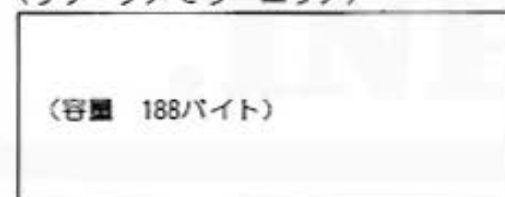
### 〈データ専用エリア〉(容量 624バイト)

Aあるいは@(1)	
Bあるいは@(2)	
Cあるいは@(3)	
...	
Zあるいは@(26)	
A\$あるいは@\$ (1)	
B\$あるいは@\$ (2)	
C\$あるいは@\$ (3)	
...	
Z\$あるいは@\$ (26)	

数値固定変数  
(26メモリ)

文字固定変数  
(26メモリ)

### 〈リザーブメモリーエリア〉



後ほど説明しますリザーブ内容、キーシンボルを記憶します。

### 〈拡張メモリー：別売〉



メモリーは上記のような構成になっています。プログラムエリアはプログラムメモリー、データメモリー(変数)のいずれにも使用することのできるメモリーです。図で示すように、プログラムと変数はそれぞれ逆方向から書き込まれ、確保されていきます。これに対してデータ専用エリアは常にデータメモリー(変数)として確保されています。通常、データメモリー(変数)として使用できるのはこの2種類のメモリーエリアですが、プログラム・データエリアを使用するときは書き込まれているプログラムの長さによって変数として使える大きさが変わってきますので注意が必要です。

次に各変数を確保する場合に使用するバイト数およびプログラムの各命令などの占めるバイト数を記しておきますので参考にしてください。



変数	変数の名前	データ	
数値変数	7 バイト	8 バイト	
文字変数	7 バイト	配列変数	指定されたバイト数※
		2 文字の変数	16 バイト



※たとえば DIM A\$(2, 3)\*10 と指定した場合は、10文字を記憶できる変数を12個確保しますが、これに使用するバイト数は7バイト(変数名)+10バイト(文字数)×12個=127バイトになります。

構成要素	ラインナンバー	命令文、関数	その他 ENTER
使用バイト数	3 バイト	2 バイト	1 バイト

注) プログラム・データエリアはプログラムとデータ用として共用しますので、プログラムの入力により、このエリアがいっぱいになり、ついには、あらかじめDIM指定などにより確保されているデータ領域を侵してしまう場合があります。

このときはデータが消され、入力したプログラムが優先して書き込まれますが、データが消去されたことを示すため、エラー(ERROR 177~181)になります。

注) プログラム・データエリアはプログラムとデータ用として共用しますので、データ領域を確保する命令を実行したときにプログラム領域を侵してしまう場合があります。このとき、通常はエラー(ERROR 10)になりますが、場合によっては  や  キーでプログラムラインを呼び出したとき、0ラインや、入力していないラインが表示されることがあります。このときは次のようにキー操作をしてください。

POKESTATUS 2-1  255 

## 5-4) 変数(メモリー)の内容を消すには(CLEAR)

○ 個々の変数をクリア(消去)するには次のようにします。

(例) A=0  
B(1)=0  
A\$=" "  
BC\$=" "

数値変数は0を代入することにより  
クリアします。

文字変数は" を代入することにより  
クリアします。

○ CLEAR命令(クリア)

すべての変数を一度にクリアするには、CLEAR命令を実行します。CLEAR命令の一般形は次のようになります。

CLEAR

●この命令ではプログラムはクリアされず、保護されます。

## 6. 表示に関するBASIC

これまでの説明では、INPUT命令やPRINT命令での表示のしかたに多くの変化はありませんでしたが、ここでは表示に多くの変化を持たせる指定のしかたや、自由な文字や記号の作りかたなどを説明します。

### 6-1) いろいろな表示のさせかた (PRINT, USING, CURSOR, WAIT, PAUSE, CLS)

#### 〔例題〕

整数 $n$ を入力し、その階乗を求めるプログラムを作ってください。

$$n! = 1 \times 2 \times 3 \times \cdots \times n$$

このとき、次のような表示をさせるようにしてください。

①プログラム実行中は ○ NO\_KAIJOU と表示させてください。

↑ 入力した数値を表示させます。

②プログラム実行後、①の表示に続いて結果を小数部5桁の指数方式で表示させてください。

〔解答例〕

```

10: INPUT "DATA_N="
   "N
20: WAIT 60          ←PRINT命令での停止時間を指定しています。
30: PRINT USING "#
   ##";N;"NO_KAIJ
   OU";
40: B=1
50: FOR A=1 TO N
60: B=B*A
70: NEXT A           ] nの階乗を計算するループです。
80: WAIT             ←PRINT命令での停止時間を無限に指定
90: PRINT "=";
   USING "#.#####" ←表示の形(フォーマット)を指定
   ^";B
100: END

```


#### 〔説明〕

##### ◎ PRINT命令

PRINT命令についてはすでに3-5)で説明しましたが、ここでもう少し説明します。  
PRINT命令には3-5)で説明しました形のほかに次の形の指定ができます。

$$\begin{array}{l}
 \text{PRINT} \left\{ \begin{array}{l} \text{式} \\ \text{"文字"} \\ \text{文字変数} \end{array} \right\}; \\
 \text{PRINT} \left\{ \begin{array}{l} \text{式} \\ \text{"文字"} \\ \text{文字変数} \end{array} \right\}; \left\{ \begin{array}{l} \text{数値変数} \\ \text{"文字"} \\ \text{文字変数} \end{array} \right\}; \cdots;
 \end{array}$$

この形の場合、PRINT命令実行により、文字や数値を表示部の左端より表示します。そして、次にPRINT命令などの表示命令がありますと、その指定内容を前の表示に続いて表示します。

<p>(例) 10:WAIT 10          20:FOR A=1 TO 26          30:PRINT "#";          40:NEXT A          50:WAIT :PRINT          60:END</p>		<p>FOR-NEXTループ          表示部の左端から右端まで          #記号を表示させます。          表示されている内容を保持したまま          プログラムの実行を停止します。</p>
---	---	--

## ◎ USING命令(ユージング)

USING命令は数値などを表示するときに、その表示フォーマットを指定します。

### ○ USING命令の一般形

- |  |  |
|--|--|
| <p>① USING "###"</p> <p>② USING "###."</p> <p>③ USING "###.##"</p> <p>④ USING "##.##^"</p> <p>⑤ USING "###.###."</p> <p>⑥ USING "+###"</p> <p>⑦ USING "*#####."</p> <p>⑧ USING "#####"</p> | <p>符号を含めて整数3桁を表示</p> <p>符号を含めて整数3桁と小数点を表示</p> <p>符号を含めて整数3桁と小数点と小数点以下2桁表示</p> <p>小数以下2桁までの指数方式での表示<br/>         [ このとき、仮数部の整数は符号を含めて2桁、指数部は符号を含めて4桁 (E-00) が自動的にとられます。 ]</p> <p>符号と3桁区切りマーク(,)を含めて整数6桁と小数点を表示<br/>         [ 数値を表示するときは、. も1桁と数えますので、たとえば-1,234,567.を表示させるときは<br/>         USING "#####.###" のように#マークを10個入れる必要があります。 ]</p> <p>正の数値に+符号をつけます。<br/>         数値は整数2桁(符号を含めて3桁)を表示します。</p> <p>符号を含めて整数5桁と小数点を表示しますが、数値の整数が指定桁より少ない場合は、少ない桁数だけ*記号をつけます。<br/>         この指定で-1を表示させた場合は -*** 1. のように表示</p> <p>文字を6桁表示</p> |
|--|--|

(注) 整数部を指定する \* と # の合計の数は次の範囲内でお使いください。

3桁区切りマーク ( , ) をつけない場合 : 11個以内

3桁区切りマーク ( , ) をつける場合 : 14個以内

本機は数値の有効桁数が10桁になっています。USING命令により整数部が10桁を超えるフォーマットを指定して、PRINT命令やLPRINT命令(プリンタ用の命令)により、数値を表示あるいは印字させたときに整数部が10桁を超えていますと正しくない数値を表示(印字)する場合があります。

(例) RUNモード

USING "#####" [ENTER] → >

PRINT 888888888888 [ENTER] → 888888888800

(LPRINT 888888888888 [ENTER] → 888888888800)

12桁

● LPRINT命令については84ページを参照ください。

- USING文を単独に使用します。

10 A=12.345, B=-34.567, C=5

20 USING "###.###"

30 PRINT A: B: C

40 END

表示例

12.34-34.56 5.00

スペース

スペース  
スペース

- PRINT文の中にUSING文を含め、PRINT USING文として使用します。

10 A=4, B=10, C=10.7703

20 PRINT "A=" : USING "###" : A:

" B=" : B: " C=" :

USING "###.###" : C

30 END

表示例

A= 4 B= 10 C= 10.770

スペース  
スペース  
スペース

スペース

スペース

スペース

スペース

- USING指定の解除

USING

USINGで表示フォーマットの指定をおこないますと、以降に実行されるPRINT



命令には、すべてそのフォーマット指定が有効になりますので、不必要なときは本命令によってフォーマット指定を解除しておきます。

```
(例) 10:A=34, B=7
      20:C=A/B
      30:USING "#####"
      40:PRINT "A=";A
      50:PRINT "B=";B
      60:PRINT USING ;" ←—— USINGによりフォーマット指定解除
          C=";C
      70:END
```

### ◎ WAIT命令 (ウェイト)

WAIT命令は、PRINT命令によるプログラムの停止時間を指定する命令です。WAIT命令による指定はPRINT命令のすべての形に対して有効で、プログラムは指定された時間停止し、その後、自動的に実行を再開します。

- 停止時間は次のように指定します。

WAIT 式

式の値により、決められている時間だけプログラム実行を停止して内容を表示し、その後プログラムの実行を再開します。

式の値は0～65535まで指定できます。なお、式の値1は約1/60秒に相当します。

- WAIT指定の解除

WAIT

WAITでの指定は、以降のPRINT命令に対してすべて有効となるため、時間を無限にしたいときなどは、本命令によって時間指定を解除してください。

### ◎ CURSOR命令 (カーソル)

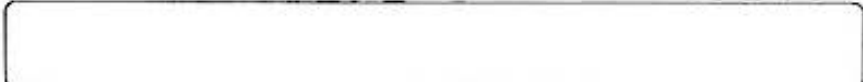
CURSOR命令はPRINT命令などにより表示される内容の表示開始位置 (ポジション) を指定します。

CURSOR 式      式の値で表示開始ポジションを指定します。

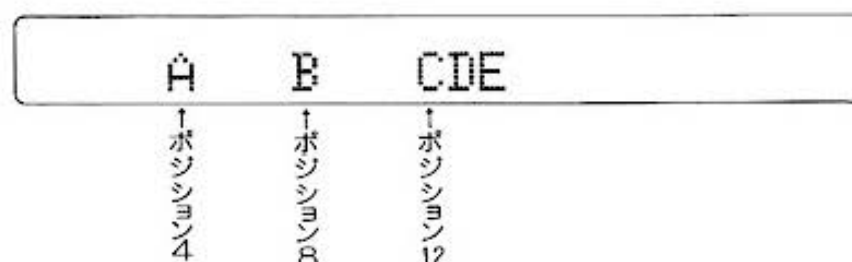
CURSOR      指定を解除します。

表示ポジションを下に示します。

ポジション→      0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25



```
(例) 10 CURSOR 4:PRINT "A"
      20 CURSOR 8:PRINT "B"
      30 CURSOR 12:PRINT "CDE"
      40 END
```



- CURSOR命令で表示開始位置を指定した場合は、以降のPRINT命令で表示された内容の前後は、それまで表示されていた内容が保持されます。この機能を使えば、表示の一部分だけを変えることもでき、応用範囲も広がります。なお、表示をクリアするときはCLS命令を用います。

#### ◎ PAUSE命令（ポーズ）

PRINT命令が表示をおこなってプログラムの実行を停止、あるいはWAIT命令により停止時間が指定されるのに対して、このPAUSE命令は停止時間が一定（約0.85秒）に決っており、その時間だけ表示をおこなって次のプログラムに実行が移るところが異っているだけで、そのほかはPRINT命令と同じです。

#### ◎ CLS命令（クリアーズ）

表示内容を消し、表示開始ポジションを0位置にもどします。

## 6-2) 文字や記号の作りかた (GCursor, GPRINT, POINT)

本機で表示される文字や記号の1つ1つは小さい点(ドット)を使って作られています。

本機内には文字や記号を表示するドットパターンのデータがたくさん記憶されていますが、このデータによらず、自由な文字や記号を作ることができれば、計算機をもっといろいろなことに活用することができます。ここでは、そのような文字や記号の作りかたについて説明します。

#### 〔例題〕

表示部の中ほどに◆マークを3個表示させるプログラムを作ってください。

#### 〔解答例1〕

```

10:GCursor &42      ← &は16進数を示す記号です。
20:WAIT 0           ← GPRINT命令での停止時間を0に指定
30:FOR A=1 TO 3
40:GPRINT "00081C   ◆マークを3個作っています。
    3E1C0800";
50:NEXT A
60:WAIT :PRINT      ← 表示を保持して、実行を停止
70:END

```

#### 〔解答例2〕

```

10:GCursor 66
20:WAIT 0
30:FOR A=1 TO 3
40:GPRINT 0;8;28;
    62;28;8;0;
50:NEXT A
60:WAIT :PRINT
70:END

```

## 〔説 明〕

## ◎ G C U R S O R 命令(グラフィックカーソル)

G C U R S O R 命令は、基本的には C U R S O R 命令と同じですが、C U R S O R 命令が桁の単位で表示開始ポジションを指定したのに対し、G C U R S O R 命令はドット(点)の列の単位で表示開始ポジションを指定します。

- 表示部はタテに7個並んだドットの列が156列あります。この156列に左側から順に0～155(& 0～& 9 B)までの番号をつけ、これにより表示開始ポジションを指定します。

(例) G C U R S O R 5      5番のポジション(左から6列目)を指定します。  
 G C U R S O R 6 2      62番のポジション(左から63列目)を指定します。  
 G C U R S O R & 3 E    16進数での3 E番(10進数での62番)のポジションを指定します。

## ◎ 16進数

16進の数値は、10進数の10～15に対する数字をA、B、C、D、E、Fと規定した数字(0～9 A B C D E F)を用いて表わします。

10進数→… 9 10 11 12 13 14 15 16 17 …… 30 31 32 ……

16進数→… 9 A B C D E F 10 11 …… 1 E 1 F 20 ……

プログラム上では10進数と16進数を区別するため、16進数値の前に&記号をつけて表わします。

(例) & 1 4      16進数の14を表わします。  
 & 2 A      16進数の2 Aを表わします。

## ◎ G P R I N T 命令(グラフィックプリント)

G P R I N T 命令はドットパターンを指定することにより、そのパターンを表示する命令で次のような形で指定します。

G P R I N T " O O O O O O …… "

↑ ↑ ↑  
 それぞれタテ1列の  
 ドットパターンを指  
 定します。

ドットの指定はタテ7ドットを上4ドット、下3ドットに分割し、それぞれを16進数字により指定します。

(例) G P R I N T " 0 C 1 2 2 2 4 7 2 2 1 2 0 C "

↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑      上側のドットパターンを表わします。  
 ↑ ↑ ↑      下側のドットパターンを表わします。

↑ ↑ ↑ ↑      4ドット  
 ↑ ↑ ↑      3ドット

0 C 2 2 7 2 2 C 0 ← 上側のドットパターンを16進数字で表わした場合。  
 0 0 1 2 4 2 1 0 0 ← 下側のドットパターンを16進数字で表わした場合。

◎ 16進数字により、それぞれ次のドットパターンが指定されます。

16進数字	ドットパターン	16進数字	ドットパターン	16進数字	ドットパターン	16進数字	ドットパターン
0		1		2		3	
4		5		6		7	
8		9		A		B	
C		D		E		F	

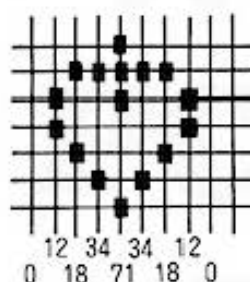
上側(4ドット)でのみこのドットパターンが指定できます。

○10進数によるドットパターン指定

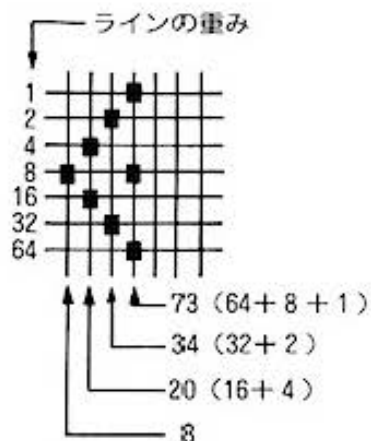
ドットパターンを10進数で指定する場合は次のようにします。

GPRINT ○:○:○:……

(例) GPRINT 12:18:34:71:34:18:12



この数値は、ドットのヨコの列(ライン)に次のような重みをつけ、表示されるドットの重みを加えることにより得ることができます。

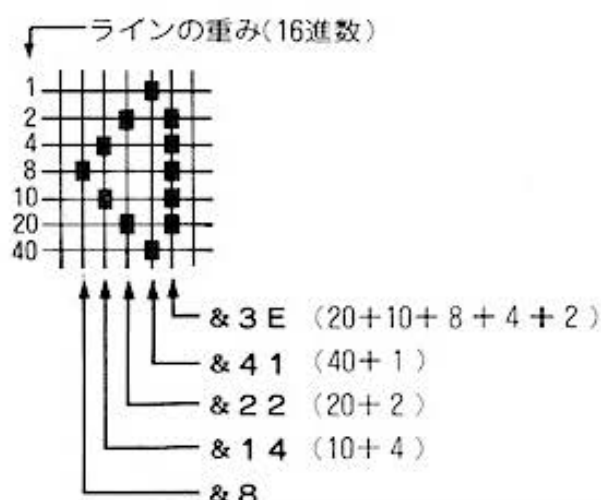


## ○16進数値によるドットパターン指定

10進数で指定する場合と同様の形で指定しますが、各数値の前に16進数を示す&マークをつけます。

(例) `GPRINT &C; &12; &22; &47; &22; &12; &C`

なお、ドットパターンの16進数値を得る場合は各ラインに次のように16進数値の重みをつけます。



## ◎ POINT命令(ポイント)

POINT命令は表示されている内容のドットパターンでの情報を読む命令で、指定したポジションに表示されているドットパターンを10進数値で読み出します。

ポジションはGCURSOR命令と同様、表示部のドットの列(7ドット×156列)に左側から順に0~155までの番号をつけ、次の形で指定します。

POINT 式

(例) `A=POINT 5`      5番のポジションのドットパターン情報を変数Aに入れます。

`B=POINT 120`      120番のポジションのドットパターン情報を変数Bに入れます。

(例) ◆を3個表示させ、表示全体を白黒反転させるプログラム。

```

10:GCURSOR 66
20:WAIT 0
30:FOR A=1TO 3
40:GPRINT "00081C
      3E1C0800";
50:NEXT A
60:WAIT 60:PRINT
70:WAIT 0
80:FOR J=1TO 4
90:FOR K=0TO 155
100:GCURSOR K
110:GPRINT (255-
      POINT K)
120:NEXT K
130:NEXT J
140:END

```

表示の白黒を  
反転させてい  
ます。

(255は  
127で  
もよい。)

## 7. そのほかのBASIC

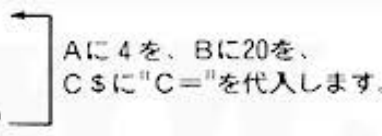
### 7-1) READとDATA、RESTORE (リードとデータ、リストア)

変数にデータを入れる(代入する)方法として、LETやINPUTのほかにREAD命令とDATA命令の組み合わせがあります。

これはREADに続いて変数を書き、そのデータをDATAに続いて書きます。この場合の一般形は次のようになります。


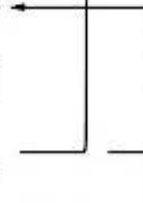
```
READ 変数,変数,.....
      ⋮
DATA 式,式,..... (DATA 文字列, 文字列,.....)
```

- DATA文の中に、式と文字列を混ぜて書くこともできます。

(例) 10:READ A, B, C\$  
20:C=A\*B  
30:PRINT C\$;C  
40:DATA 4, 20, "C="   
50:END

Aに4を、Bに20を、  
C\$に"C="を代入します。

- 上例のようにREADの後の変数と、DATAの後のデータは1対1で対応します。したがって変数の形とデータの形が一致していなければなりません。すなわち、数値変数に対応するデータは数値でなければなりませんし、文字変数に対応するデータは文字列でなければなりません。
- 1つのプログラムの中に何回でもREAD文およびDATA文を書くことができますが、データは何ラインにわけて書いても一連のデータと見なされ、若いラインのデータから順番に変数に代入されます。

(例) 10:READ A, B   
20:G=J(A\*A+B\*B)  
30:READ C, D, E\$, F\$   
40:H=C\*COS D  
50:PRINT E\$;G;F\$;  
H  
60:DATA 3, 4, 5  
70:DATA 60, "G=", "  
H=" "  
80:END

Aには3、Bには4が代入されます。

Cには5、Dには60、  
E\$には"G="、F\$  
には" H="、が代  
入されます。

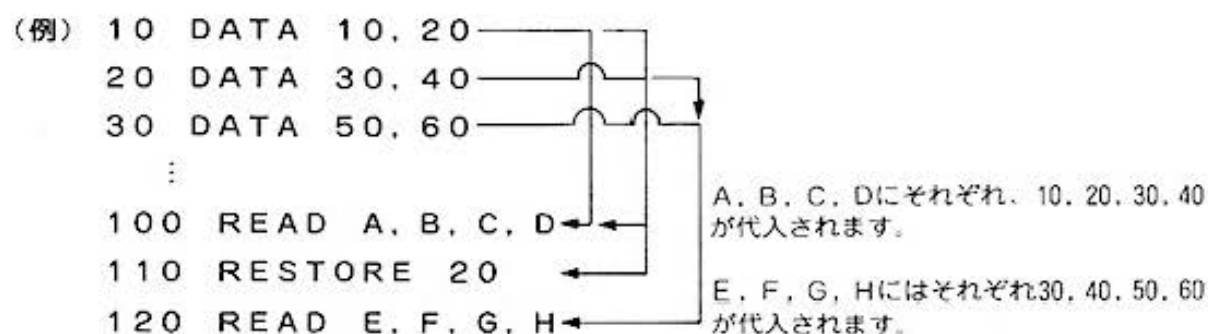
#### ◎ RESTORE命令 (リストア)

READ命令の実行時、DATA命令で指定されているデータのどのデータを読み込むかは常に計算機に記憶されていますが、この読み込むデータの順番を強制的に変えるときに使用します。



## ○ RESTORE 式

この形の場合、この命令を実行した次のREAD命令の変数には、RESTORE命令の式の値で指定されるラインに書かれているDATA命令の最初のデータから1対1に対応して代入されます。式の値はRESTORE命令が書かれているラインより必ず小さくしてください。



## ○ RESTORE

この形の場合、この命令を実行した次のREAD命令の変数には、プログラムの最も若いラインに書かれているDATA命令のデータが1対1に対応して代入されます。

## 7-2) ON GOTO, ON GOSUB

## ◎ ON GOTO命令(オン・ゴートゥ)

GOTO命令についてはすでに述べましたが、ON GOTO命令はGOTO命令の機能を拡大したものです。ON GOTO命令の一般形は次のようになります。

ON 式 GOTO 式1, 式2, 式3, …… (ON式 GOTO "ラベル", "ラベル", ……)

この命令ではONの次の式の値が1の場合はGOTOに続いて書かれている式1で示されるラインへジャンプし、ONの次の式の値が2の場合は式2で示されるラインへとジャンプします。もし、式の値が、1より小さいときやGOTOに続いて書かれている式の数より大きい値のときは、ON GOTO命令の書かれている次のラインへ実行が移ります。

(例)

```

10: INPUT A
20: ON AGOTO 110, 1 ← Aの値が1のときは110ラインへ、
    20, 130           2のときは120ラインへ、3のとき
    30: B=4, C=64      は130ラインへジャンプします。
40: CLS : GCURSOR C   Aの値が1より小さく、3より大
50: WAIT 0            きは次のラインへ実行が移
60: FOR A=1 TO B       ります。
70: GPRINT "00081C
    3E1C0000";
80: NEXT A
90: WAIT : PRINT
100: END
110: B=1, C=74: GOTO
    40
120: B=2, C=71: GOTO
    40
130: B=3, C=67: GOTO
    40

```

注) GOTOに続く式には、二次元配列変数(A(1,2), B\$(5,3)など)を使用することはできません。

#### ◎ ON GOSUB命令(オン・ゴーサブ)

ON GOTO命令と同じ形で指定すれば、ONに続く式の値により、GOSUBに続いて書かれた式で示すラインへサブルーチンジャンプします。

## 7-3) ON ERROR (エラー処理) (オン・エラー)

プログラムの実行中はいろいろなエラーが発生する可能性があります。エラーが発生すると計算機はエラーコードを表示して実行が止ってしまいますので、このようなときに、エラー処理をおこなうためにこのON ERROR命令を用います。

ON ERROR GOTO 式

プログラムの中でON ERRORを宣言しておきますと、以降の実行で、もしエラーが発生してもプログラムの実行は停止せずに、GOTO命令で指示されたラインへジャンプします。(文法エラー(エラーコード1番)およびINPUT命令実行時に発生するエラー(エラーコード7番、32番など)は除きます)そこでエラー処理をおこなって復帰させてやるのが通常の使い方です。

```
(例) 10:ON ERROR GOTO
      50
      20:INPUT "X=";X
      30:PRINT "√X=";√X ← Xの値が負数のときはエラーに
      40:GOTO 20          なり、50ラインへジャンプします。
      50:PRINT "HUSUU_W
      A_KEISAN_DEKIM
      ASEN"
      60:GOTO 20
      70:END
```

## 7-4) REM(リマーク)

この命令は、プログラムの実行には関係なく、プログラムリストなどをわかりやすくする目的でプログラムの先頭や途中に注釈を入れておくものです。

```
(例) 10 REM KINRI_KEISAN
      ⋮
      200 REM SUBROUTINE
      ⋮
```

## 7-5) ARUN (オートラン)

この命令は電源を入れたとき、自動的にプログラムの実行を開始させる命令で、プログラムの最初にこの命令を書き込んでおけば、**OFF** キーにより切られた電源を **ON** キーにより入れたときに、自動的にプログラムがスタートします。ただし、**ON** キーを押したときにモードがRUNモードになっていることが必要です。そしてオプション機器に異常がないことが必要です。

注) **OFF** キーにより電源が切られているときにオプション機器（プリンタ・カセットインターフェイス、拡張メモリーなど）をつけたり、はずしたり、またスイッチなどを切替えたりするなど、状態を変えますと、計算機はオプションに異常があると受け取ってオートランしないことがあります。このような場合はRUN命令などでプログラムを実行してください。



```
(例) 10: ARUN : WAIT 128
      20: PRINT "KINRI_K
        EISAN"
      30: INPUT "GANKIN_
        WA "; A, "KINRI_
        WA "; B, "KIKAN_
        WA "; C
      40: D=A*B/100*C/36
        5
      50: WAIT : PRINT "R
        ISOKU_WA "; D
      60: END
```

○オートランによるプログラムの実行開始時はプログラムの実行中を示すBUSYシンボルが表示されません。

○オートランによりプログラムの実行を開始したときは、変数や各種指定（ユージング指定、カーソル指定、ウェイト指定など）は消去および解除されませんので必要に応じてCLEAR命令での変数の消去や、それぞれの命令で指定の解除などをおこなうようにプログラムしてください。

## 7-6) STOP (ストップ)

この命令はプログラムの一時停止命令です。この命令を実行しますと、「BREAK IN 200」のように、実行したラインナンバーを持ったブレークメッセージを表示して実行を停止します。

このとき、、などのキーを押しますと、STOP文実行前の表示内容呼び出すことができます。

```
(例) 10: INPUT A, B, C, D
      20: E=A*COS B-C*
        COS D
      30: F=C*SIN D-A*
        SIN B
      40: STOP
      50: G=J(E*E+F*F)
      60: H=J*(G/2)^2
      70: PRINT G
      80: PRINT H
      90: END
```

STOP命令により実行が一時停止している状態(ブレーク状態)のときはマニュアル操作が可能になるため、変数の状態を調べたり、変数に数値を代入したりする場合によくこの命令が使用されます。

○STOP命令により停止した実行を再開させるときは CONT ENTER と操作してください。

## 7-7) AREAD (オートリード)

この命令は定義付キーにより、プログラムの実行を開始したとき、表示されていた内容を指定された変数に読み込む命令で、定義付プログラム(39ページ参照)で、ラベルに続いて書きます。この命令の一般形は次のようになります。

AREAD 変数

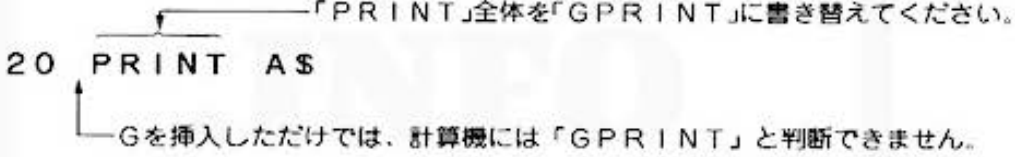
(例) 10 "A": AREAD A  
:  
100 "B": AREAD A\$

○この命令は、プログラム実行開始直後以外はスキップ(無視)されます。

○プログラムの実行を開始したとき、表示がプロンプト表示になっていたときは、指定されている変数はクリアされます。

### ご注意

プログラムに書き込まれているPRINT命令をGPRINT命令に訂正するときは次のことに留意してください。

(例)   
20 PRINT A\$  
Gを挿入しただけでは、計算機には「GPRINT」と判断できません。  
(「G」と「PRINT」と見なします。)

これはCURSOR命令をGCURSOR命令に訂正する場合や、プリンタ関係の命令についても同様です。

## 8. 特別な機能

### 8-1) 乱数関数(RND, RANDOM)

#### ◎ RND命令(ランダム)

RND命令は乱数を発生させる機能を持つ関数です。

RND X においてXの値により次のような乱数を得ることができます。

- Xが負数の場合： 直前に発生した乱数(あるいは乱数列)と同じ乱数を発生させるために、初期値を一定にします。
- Xが0から1未満の場合： 0から1未満の乱数を発生します。
- Xが1以上の場合： 1からXの値以下の乱数を発生します。なお、このときの乱数は整数値になります。

#### (例1) RUNモードでのマニュアル操作の場合

```
RND 0 [ENTER] 1.902513749E-01
RND 0 [ENTER] 3.757816235E-01
RND 0 [ENTER] 6.429773406E-01
```

#### (例2) プログラム例

```
10:WAIT 0
20:FOR A=1TO 3
30:D=RND -1 ←この代入文は同じパターンの乱数を発生さ
40:FOR B=1TO 4   せるために入れています。
50:C=RND 9       このラインを削除すれば同じパターンの乱
60:PRINT C;      数は発生されなくなります。
70:NEXT B        (変数Dは、Dにかぎらず、プログラム内で
80:NEXT A        他に使用していない変数であればよい。)
90:WAIT :PRINT
100:END
```

表示例

```
9 7 1 3 9 7 1 3 9 7 1 3
```

4個の乱数を3回表示させています。

○乱数の有効桁数は10桁です。

#### ◎ RANDOM命令(ランダムイズ)

上記RND命令の使用に先立って乱数のタネを植えつけるものです。

RNDは電源オンからやり直すと、常に同じ乱数を発生します。しかし、電源オン後にRANDOM命令を実行すれば同じ乱数を発生することは、まず、なくなります。

## 8-2) TIME (タイム)

TIMEは月日時分秒のカレンダー時計を表わす特殊な数値変数です。

○カレンダー時計の設定方法

TIME = 20810  3205   
                   2月8日10時      32分5秒

○カレンダー時計の呼び出し方法

TIME

注) このカレンダー時計は一度設定すれば、 キーなどで電源を切っても動作している24時間制の時計で、月・日も正しくカウントします。(ただし、うるう年の判断機能は持っていません。)

○TIMEは変数と同じ扱いができますので、式の中などでも自由に使用することができます。

## 8-3) INKEY\$ (インキードル)

プログラムの中で、この命令が実行されたとき、いずれかのキーが押されていれば、その内容を読み込んで指定された変数に代入します。

INKEY\$の一般形は次のようになります。

文字変数 = INKEY\$

(例) 10:RANDOM  
 20:WAIT 0:CURSOR  
       4  
 30:PRINT "MARU\_KA  
       \_BATU\_KA\_?";  
 40:A=RND 2  
 50:A\$=INKEY\$:IF  
       A\$="A"THEN 70  
 60:GOTO 40  
 70:CLS:GOCURSOR 7  
       4  
 80:WAIT 120  
 90:IF A=2THEN 120  
 100:GPRINT "1C2241  
       4141221C"  
 110:GOTO 20  
 120:GPRINT "412214  
       08142241"  
 130:GOTO 20

このラインをくり返し実行し、 キーが押されるのを待ちます。

〔このプログラムは、実行開始後  キーを押せば○か×を表示するプログラムです。  
 プログラムの実行を終了するときは   と押してください。〕



注1) プログラムのはじめに `INKEY$` がありますと、`RUN` ENTER などプログラムをスタートさせたとき、ENTER キーを読み取ってしまうことがありますので注意が必要です。

注2) ソフトウェアキーが、`INKEY$`により読み込まれた場合はアスキーコードで17~22に相当する文字(記号)が読み込まれます。(、"、#、\$、%、&ではありません)ただし、本機の場合はキャラクタコード17~22には表示できる文字や記号はありません。したがって、先のプログラムの A キーの替りに、たとえば I キーを用いる場合は、`CHR$`関数を使って、

```
50 A$=INKEY$: IF A$=CHR$ 17 THEN 70
```

のように指定します。(CHR\$関数については74ページを参照してください。)

注3) `INKEY$`命令は SHIFT キーや DEF キーが押されていれば、そのキーを読み込みます。したがって、SHIFT キーに続いて押したときに働く機能や入力される文字、DEF キーに、続いて押したときに働く機能などを読み込むことはできません。

## 8-4) 音発生機能

### (BEEP, BEEP ON, BEEP OFF)

本機には音発生機能がついていますので、プログラムの途中で音を出し、計算機の操作をより効果的に、また楽しくすることができます。

#### ◎ BEEP命令(ビープ)

BEEP命令は音の発生回数、音の高さ、長さを指定する命令です。BEEP命令の一般形は次のようになります。

BEEP 式1, 式2, 式3

この一般形の式1、式2、式3により、それぞれ次の指定をおこないます。

式1: 音の発生回数を指定します。式1の値は0~65535の範囲内にしてください。

式2: 音の高さを指定します。式2の値を255~0にすることにより、約230Hz~約7 kHzの周波数の音を指定することができます。

式3: 音の長さを指定します。式3の値は0~65279の範囲内で指定してください。なお、音の長さは式2の値(周波数)により、大きく変化します。

(式2の値が大きくなる(周波数が低くなる)と、式3の値が同じでも音が出ている時間が長くなります。)

- 式3あるいは式2と式3を省略することができます。式3を省略した場合は、その値が160に指定され、式2を省略した場合は、音の高さ(周波数)が約4 kHzに指定されます。

#### ◎ BEEP ON命令(ビープオン)

この命令は音の発生機能をオンにする命令です。この命令実行後のBEEP命令では指定された音が出ます。また、カセットの記録・再生音も発生できるようになります。

#### ◎ BEEP OFF命令(ビープオフ)

この命令は音の発生を禁止する命令です。この命令実行後はBEEP命令による音も、カセットの記録・再生音も発生されません。

(例) 10:RANDOM :BEEP  
ON  
20:WAIT 0:CURSOR  
4  
30:PRINT "MARU\_KA  
\_BATU\_KA\_?";  
40:A=RND 2  
50:A\$=INKEY\$ :IF  
A\$="A"THEN 70  
60:GOTO 40  
70:CLS :GCURSOR 7  
4  
80:WAIT 120  
90:IF A=2THEN 130  
100:BEEP 2, 70, 50  
110:GPRINT "1C2241  
4141221C"  
120:GOTO 20  
130:BEEP 4, 140, 25  
140:GPRINT "412214  
08142241"  
150:GOTO 20

(このプログラムは61ページのプログラムに  
音発生命令を追加したものです。)

注) BEEP命令は、式の値の組み合わせによっては発生する音の音質が変化することがあります。

## 8-5) CONT(コンティニュー)

CONT命令は、プログラムの実行が一時停止しているときに、プログラムの実行を再開させるための命令です。したがって、この命令はプログラムに書き込むことはできず、RUNモードでのマニュアル操作でのみ有効となります。この命令の一般形は次のようになります。

CONT ENTER

○プログラムの一時停止状態(ブレーク状態)とは次の状態です。

- ①プログラム実行中にSTOP命令で一時停止した状態
- ②プログラム実行中に BREAK  
ON キーを押して一時停止した状態
- ③そのほか、PRINT命令の実行などにより実行が停止している状態

## 8-6) STATUS、MEM(ステータス、メモ)

プログラムの長さや、プログラム・データメモリーの中でプログラム以外の部分のバイト数などを確認するための命令です。

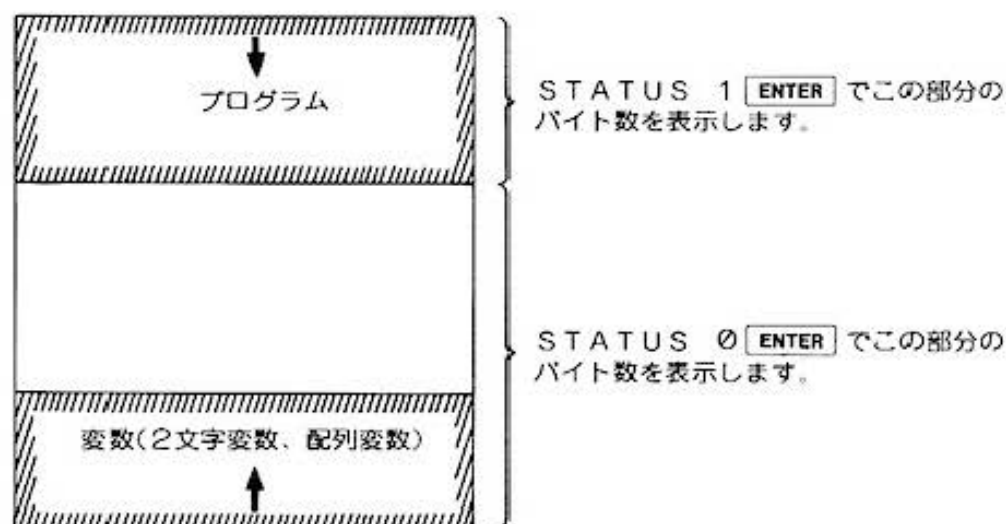
◎ STATUS 0 ENTER

この命令を実行すれば、プログラム・データメモリーのプログラムが書き込まれていない部分のバイト数を表示します。(バイトとは計算機のメモリー容量を表わす単位です。)

◎ STATUS` 1 ENTER

この命令を実行すれば、書き込まれているプログラムのバイト数を表示します。

# プログラム・データメモリー



## ◎ MEM **ENTER**

この命令はSTATUS 0 **ENTER** と同じ働きをします。

## 8-7) 論理積、論理和、否定(AND, OR, NOT)

### ◎ 論理積(AND: アンド)

論理積は次のような値を取る関数です。

$$1 \text{ AND } 1 = 1$$

$$1 \text{ AND } 0 = 0$$

$$0 \text{ AND } 1 = 0$$

$$0 \text{ AND } 0 = 0$$

たとえば41と27の論理積(AND)を計算する場合は、これらの数値を2進数に変換して、それぞれの桁のANDを取り、その結果を10進数に変換します。

$$41 \text{ AND } 27 = 9$$

$$\begin{array}{rcl} \text{AND} < \begin{array}{l} 101001 \text{ --- } 41 \\ 011011 \text{ --- } 27 \\ \hline 001001 \text{ --- } 9 \end{array} \end{array}$$

### ◎ 論理和(OR: オア)

論理和は次のような値を取る関数です。

$$1 \text{ OR } 1 = 1$$

$$1 \text{ OR } 0 = 1$$

$$0 \text{ OR } 1 = 1$$

$$0 \text{ OR } 0 = 0$$

たとえば41と27の論理和(OR)はAND命令と同様、2進数に変換して、それぞれの桁のORを取りその結果を10進数に変換します。

$$41 \text{ OR } 27 = 59$$

$$\begin{array}{r} \text{OR} \left\{ \begin{array}{l} 101001 \text{ --- } 41 \\ 011011 \text{ --- } 27 \\ \hline 111011 \text{ --- } 59 \end{array} \right. \end{array}$$

## ◎ 否定(NOT:ノット)

10進数を16桁の2進で表わすと次のようになります。

10進	16桁の2進
32767	0111111111111111
⋮	⋮
3	0000000000000011
2	0000000000000010
1	0000000000000001
0	0000000000000000
-1	1111111111111111
-2	1111111111111110
-3	1111111111111101
⋮	⋮
-32768	1000000000000000

ここで2進数0000000000000001の否定(NOT)を取りますと次のようになります。

$$\begin{array}{l} \text{NOT (否定)} \left\{ \begin{array}{l} 0000000000000001 \\ 1111111111111110 \end{array} \right. \end{array}$$

このように、各桁の1を0に、0を1に反転することを“否定(NOT)を取る”といいます。

この数値(1111111111111110)を10進数で表わす場合

$$\begin{array}{r} \text{(2進数)} \quad \text{(10進数)} \\ 1111111111111110 = ? \\ +) 0000000000000010 = 2 \\ \hline 0 = 0 \end{array}$$

の関係から?は-2であることがわかります。さらに

$$\text{NOT } 0 = -1$$

$$\text{NOT } -1 = 0$$

$$\text{NOT } -2 = 1$$

となり、一般的に数値Xの否定は  $\text{NOT } X = -(X + 1)$  の関係式で表わせます。

## ＝ご注意＝

AND、OR、NOT は計算機内部では2進数で求められていますが、この結果に対して、さらにAND、OR、NOTを実行すると正しく動作しない場合がありますので、ご注意ください。

例  $\frac{A}{10\text{進数}} \text{ OR } \frac{(\text{NOT } 2)}{2\text{進数}}$

このような場合は間に--(符号)を入れ、2進数を10進数に変換した上でお使いください。

$\frac{A}{10\text{進数}} \text{ OR } \frac{--(\text{NOT } 2)}{2\text{進数}}$   
10進数

## 8-8) 条件式の結合 (AND, OR)

これまで、IF文の中で使用しました条件式は

$A > 5$

$A\$ = "A"$

$A * B \geq C + 20$

などのような、2つの値の大小比較の式が1つのみでしたが、これらの大小比較の式を2つ以上結合して、1つの条件式をつくることができます。たとえば“Aは5よりも大きくかつ10よりも小さい”という条件を式で表わす場合、これまではIF文を2つ使って

```
10 IF A<5 THEN 30
20 IF A<10 THEN 60
30
...
```

のような形で表わす必要がありましたが、AND命令を使えば

```
10 IF A>5 AND A<10 THEN 60
```

のように1つの条件式にすることができます。

## ◎ AND命令(アンド)

2つ以上の条件のすべてを満足するような条件を1つの式で表わすときに用います。

(例) Aは0よりも大きくかつ6より小さい場合の条件式

$A > 0 \text{ AND } A < 6$

## ◎ OR命令(オア)

2つ以上の条件のいずれかを満足するような条件を1つの式で表わすときに用います。

(例) Aは0よりも小さいか、あるいは12より大きい場合の条件式

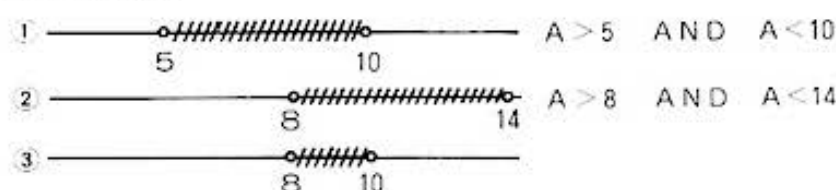
$A < 0 \text{ OR } A > 12$

○AND, OR, を使った条件式の例

①  $A > 5$  かつ  $A < 10$

②  $A > 8$  かつ  $A < 14$

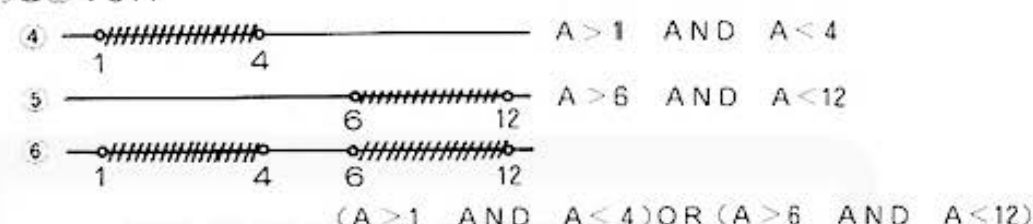
③ ①と②のAND



④  $A > 1$  かつ  $A < 4$

⑤  $A > 6$  かつ  $A < 12$

⑥ ④と⑤のOR



＝参 考＝

論理計算は次のように構成することにより、論理和 (OR)、論理積 (AND) を形づくることができます。

① 論理和 (論理計算) + (論理計算)

例  $(A < 0) + (A > 8)$   $A$  が 0 より小さいか、あるいは 8 より大きいときに 1 の値を取る  
 $(B > 0) + (C > 0)$   $B$  あるいは  $C$  が 0 より大きいとき、および  $B$ 、 $C$  ともに 0 より大きいとき 1 (あるいは 2) の値を取る

② 論理積 (論理計算) \* (論理計算)

例  $(B > 1) * (B > 6)$   $B$  が 1 より大きく、かつ 6 より小さいとき 1 の値を取る

③ 特殊な使い方の例

例  $B = (A = 1) * 3 + (A = 2) * 7$   $A$  が 1 のとき  $B$  は 3 になり  $A$  が 2 のとき  $B$  は 7 になるその他のときは  $B$  は 0 になる

## 8-9) モードロック (LOCK, UNLOCK)

◎ LOCK 命令 (ロック)

**MODE** キーをロックして、モード (RUN, PRO, RESERVE) を他に移動させなくする命令です。たとえば RUN モードにロックすれば、**MODE** キーを押しても PRO モードに移動することがありませんので、誤って大事なプログラムをこわしたり、消したりすることがありません。この命令の一般形は次のようになります。

LOCK **ENTER**

◎ UNLOCK 命令 (アンロック)

LOCK 命令を解除する命令で一般形は次のようになります。

UNLOCK **ENTER**



## 9. デバッグ(TRACE、TR OFF)

プログラムを実行した場合に、何かの誤りで迷走したり、思わぬ結果が出てきたりすることがありますが、ここではこのようなとき、1ラインずつ経過をたどりながらプログラムを実行する方法について説明します。(このような方法をデバッグといいます。)

### ◎ デバッグのしかた

- (1) デバッグはRUNモードでおこないますので、RUNモードにしてください。
- (2) TRACE ON  と押し、トレースモードを指定します。
- (3) RUN  と押してプログラムの実行を開始します。(最初のラインの実行が終われば実行は停止します。)
- (4) その後は  キーを押します。(1ラインだけ実行して停止します。)ただし、INPUT命令によるデータの入力や、PRINT命令により停止しているときの再開は、通常のプログラムの実行と同じように  キーでおこないます。
- (5) デバッグが終了すれば TRACE OFF  と押してトレースモードを解除してください。

次にごく簡単な例を示します。

#### 〈プログラム例〉

```
10 INPUT "A="; A, "B="; B
20 C=A*2
30 D=B*3
40 PRINT "C="; C; " _D="; D
50 END
```

#### 〈実行〉

RUNモード

TRACE ON	<input type="button" value="ENTER"/>	>	
RUN	<input type="button" value="ENTER"/>	A= _	} INPUT命令実行
8	<input type="button" value="ENTER"/> (データ入力)	B= _	
9	<input type="button" value="ENTER"/> (データ入力)	10:	←10ライン終了
	<input type="button" value="↓"/>	20:	←20ライン終了
	<input type="button" value="↓"/>	30:	←30ライン終了
	<input type="button" value="↓"/>	C=16 D=27	←PRINT命令実行
	<input type="button" value="ENTER"/>	40:	←40ライン終了
	<input type="button" value="↓"/>	>	←実行終了

○デバッグ動作でラインナンバーを表示しているときは、変数の内容をマニュアル操作によって呼び出し、所望の値になっているかどうかチェックできます。また、このとき  を押せば、押している間止まっているラインの内容を表示します。(なお、 キーを押して離れたときは、INPUT命令や、PRINT命令などで表示されていた内容が表示されます。)

注) トレースモードは TR OFF **ENTER** と押すが、**SHIFT** **CL**<sup>CA</sup> と押すまで指定された状態が保たれます。

### ◎ プログラムの途中で実行を停止させてチェックする場合

○プログラムの中にSTOP命令を書いておけば、プログラム実行途中でSTOP命令を実行し、ブレークメッセージを表示して実行が停止します。

このとき

- ① マニュアル操作で変数の内容をチェックする。
- ② トレースモードを指定して **↓** キーの操作にて以降のラインを1ラインずつデバッグ操作する。
- ③ トレースモードを解除して CONT **ENTER** と操作し、通常の実行状態にて続行する。

○通常のプログラム実行中に **BREAK ON** キーを押しますと、現在実行しているラインの終わりで実行を停止し、ブレークメッセージを表示します。このときも前記同様3つの操作をおこなうのが普通です。なお、ブレーク状態(停止状態)のときに **↑** を押せば、押している間停止しているプログラムラインが表示されます。

(なお、**↑** を離したときはINPUT命令や、PRINT命令などで表示されていた内容が表示されます。)

○トレースモードを指定しない場合でも、**BREAK ON** キーなどにより一時停止しているプログラムを **↓** キーにより1ラインずつ実行させていくことができます。ただし、このときはラインナンバーは表示されず、INPUTやPRINTなどの命令により表示されていた内容がそのまま表示されます。

## 10. 文字の編集

この項では、文字を分断したり、結合したり、また、文字列の大小比較など、文字と文字の扱いについて説明します。

### 10-1) 文字列の分断 (LEFT\$, RIGHT\$, MID\$)

#### ◎ LEFT\$ 関数 (レフトドル)

この関数は、ある文字列の左側から何文字かを取り出す関数です。この関数の一般形は次のようになります。

LEFT\$ (文字変数, 数式)

あるいは

LEFT\$ ("文字", 数式)

たとえば A\$ = "ABCDE" のとき LEFT\$ (A\$, 3) は A\$ の文字列の左側 3 文字、すなわち "ABC" を取り出しなさい" という意味になります。

```
(例) 10: A$ = "POCKET_COM
      PUTER"
      20: WAIT 30
      30: FOR H=1 TO 15
      40: B$ = LEFT$ (A$, H)
      50: CURSOR 0
      60: PRINT B$;
      70: NEXT H
      80: WAIT : PRINT
      90: END
```

このプログラムを実行すれば表示は

```
P
PO
POC
...
POCKET COMPUTER
```

のように左の文字から順番に表われてきます。

#### ◎ RIGHT\$ (ライトドル)

この関数は LEFT\$ 関数とは逆に、右側から何文字かを取り出す関数です。一般形は次のようになります。

RIGHT\$ (文字変数, 数式)

あるいは

RIGHT\$ ("文字", 数式)

(例) 先の例のプログラムの40ラインを次のように書きかえて実行してみてください。

```
40 B$=RIGHT$(A$, H)
```

この場合は

```
R
ER
TER
...
POCKET COMPUTER
```

のように右の文字から表われ、順順に右に送られていきます。

### ◎ MID\$(ミッドドル)

この関数は文字列の中間の文字だけを取り出す関数です。この関数の一般形は次のようになります。

↓ 左側何文字目から取り出すか指定  
 MID\$(文字変数, 数式1, 数式2)  
 ↑ 何文字を取り出すか指定  
 MID\$("文字", 数式1, 数式2)

たとえば A\$="ABCDE" のとき、MID\$(A\$, 2, 3)と指定した場合は、A\$の左側2文字目から3文字、すなわち"BCDを取り出しなさい"という意味になります。

```
(例) 10:A$="POCKET_COM
      PUTER"
      20:B=1
      30:FOR H=1 TO 8
      40:B$=MID$(A$, 9-
        H, B)
      50:B=B+2
      60:CURSOR 13-H
      70:WAIT 70:PRINT
        B$;
      80:NEXT H
      90:WAIT :PRINT
     100:END
```

このプログラムを実行すると

```
      C
      CO
      T COM
      ...
POCKET COMPUTER
```

のように真中の文字から表われます。

## 10-2) 文字列の結合

いくつかの文字列を結合させて、新しい文字列を作る場合は、“文字列の加算”をおこないます。

たとえば、A\$="ABC"、B\$="DEF"、“XYZ”の3つの文字列を結合させる場合は

A\$+B\$+"XYZ"

のように加算の形で指定します。

```
(例) 10: READ A$, B$, C$,
      D$
      20: E$=A$+C$+B$
      30: F$=D$+B$
      40: WAIT 60: PRINT
          E$;
      50: WAIT 100: PRINT
          " _ " + F$
      60: DATA "AREWA _",
          " _ ?"
      70: DATA "INU", "NE
          KO"
      80: END
```

注) 通常文字変数は16文字までを記憶することができます。文字列を変数に代入する場合は16文字を超えないようにしてください。もし、16文字を超えるような場合は文字配列変数により文字数を指定(DIM指定)してお使いください。

## 10-3) 文字数を数える方法(LEN)

1つの文字列の中に含まれる文字の数(記号、スペース、数字も含みます)は計算機により、たちどころに求めることができます。

### ◎ LEN関数(レングス)

たとえば




A=LEN "PC-1500"

のように指定して実行すれば "PC-1500" の文字数 7が変数Aに代入されます。この関数の一般形は次のようになります。

LEN { "文字" }  
      文字変数

```
(例) 10: INPUT "ANATA_N
      O_NAMAE_WA?",
      A$
      20: N=LEN A$
      30: FOR H=1 TO N
      40: B$=LEFT$ (A$, H
          )
      50: CLS : CURSOR 0
      60: WAIT 12: PRINT
          B$
      70: NEXT H
      80: GOTO 30
```

● 名前を入れれば、名前の頭文字から順番に1字ずつ表示させるプログラムです。

(プログラムの実行を終わるときは    と押します。)

## 10-4) 数値と文字の変換 (VAL, STR\$)

数値を文字列として扱いたい場合や、逆に文字列として指定されている数字を数値として扱いたい場合がありますが、このようなとき、STR\$あるいはVAL関数を使います。

### ◎ STR\$関数(ストリングドル)

この関数は数値を文字列の形に変換する関数です。たとえば、B=1234 のとき、

A\$=STR\$ B

として実行すれば、A\$には"1234"という文字列が代入されます。この関数の一般形は次のようになります。

STR\$ 数式

(例) A\$=STR\$120

A\$="120"と同じになります。

B\$=STR\$(1.2\*3)

B\$="3.6"と同じになります。

### ◎ VAL関数(バリュー)

VAL関数はSTR\$関数とは逆に文字列を数値に変換する関数です。たとえば

B\$="1234"のとき

A=VAL B\$

として実行すればAには1234という数値が代入されます。この関数の一般形は次のようになります。

VAL { "文字"  
文字変数 }

(例) A=VAL "120"

Aに120が代入されます。

B=VAL "3.2\*4="

Bに3.2が代入されます。

VAL関数により数値に変換できる文字列は数値を表わすことのできる数字(0~9)、符号(+,-)、指数部を示す記号(E)で構成されている文字列で、他の文字や記号を含んでいないことが必要です。もし、1つの文字列の中に他の文字や記号が含まれている場合は、それから右の文字列は無視されます。ただしスペースが入っていても、それはないものと見なされます。

## 10-5) アスキーコードの変換 (ASC, CHR\$)

### ◎ アスキー(ASCII)コード

アルファベット、数字、記号などを計算機が記憶したり、処理したりする場合は、すべて計算機が取り扱いやすい数値に変換します。たとえばアルファベットのAは計算機内では65 (10進数)という数値(コード)になっています。(実際には2進数の01000001となっています。)同様にBは66、Cは67というようにコードを決めています。このコードの決めかたに何種類かあって、その1つがアスキーコード(ASCII code)です。



### ◎ ASC関数(アスキー)

この関数は文字や記号、数字などをアスキーコードに変換する関数です。たとえば「Z」という文字のアスキーコードを知りたい場合は

A=ASC"Z"

として実行すれば、Aには「Z」のアスキーコードが10進数の90として代入されます。この関数の一般形は次のようになります。

ASC { "文字"  
文字変数 }

なお、文字列が2文字以上指定された場合は、頭の文字のみがアスキーコードに変換されます。

### ◎ CHR\$関数(キャラクタドル)

この関数はASC命令とは逆の関数で、10進数のアスキーコードを文字や記号(0～9の数字を含みます。)に変換する関数です。たとえば、アスキーコード「90」の文字を知りたい場合は、

A\$=CHR\$ 90

として実行すれば、A\$には「Z」が代入されます。CHR\$の一般形は次のようになります。

CHR\$ 数値

なお、本機でのアスキーコードは0～255までありますが、文字や記号(これらを総称してキャラクタ:Characterといいます)に対応しているコードは32～127(160～255)までです。(169ページ参照)

```
10:FOR H=32TO 127
20:WAIT 50:CURSOR
  8
30:PRINT H;"_ _ _";
  CHR$ H
40:NEXT H
50:END
```

●このプログラムはアスキーコード32～127に対応するキャラクタを表示させるプログラムです。(ただしコード32はスペースです。)

## 10-6) 文字列の比較

2つの文字列を比較して、その大小判断をおこなうことができます。(アスキーコードの大きさによる大小判断)この判断を用いれば、たとえば文字列をアルファベット順に並べ替えることも容易になります。文字列の比較は、数値の比較に用いました等号、不等号、大小比較の記号を用いることによりおこないます。たとえば

"ABC"<"BC"

"AKIRA">"AKIO"

"YAMATO"="YAMATO"

(アスキーコードではAが65、Bが66、Cが67……と  
なっています。したがって、文字列の比較では、A  
はBより小さく、BはCよりも小さくなります。ま  
た、数字はアルファベットよりも小さくなります。)

のように書き表わすことができます。いま、文字列をA\$とB\$とした場合、次の種類の比較ができます。

A\$=B\$     A\$とB\$は同じ文字列です。

A\$<B\$     A\$はB\$よりも小さい文字列です。

A\$ > B\$    A\$はB\$よりも大きい文字列です。

A\$ < > B\$    A\$とB\$は同じ文字列ではありません。

これらはIF文の中に使用して大小判断に用いることができます。

注) 文字列の比較ではA\$ <= B\$, A\$ >= B\$の形は使えません。この場合は(A\$ < B\$) OR (A\$ = B\$), (A\$ > B\$) OR (A\$ = B\$)のような形で比較することができます。

(例) 5人の名前を入れて、アルファベット順に並べ替えるプログラム。

```

10: DIM A$(4)*10
20: FOR H=1 TO 5
30: WAIT 0: CLS
40: PRINT "NAME";
   H; "____";
50: INPUT A$(H-1)
60: NEXT H
70: FOR B=0 TO 3
80: FOR C=B+1 TO 4
90: IF A$(B)=A$(C)
   THEN 120
100: IF A$(B)<A$(C)
   THEN 120
110: D$=A$(B): A$(B)
   =A$(C): A$(C)=D$
120: NEXT C
130: NEXT B
140: "A": FOR J=1 TO
   5
150: WAIT 140: CLS
160: PRINT "NO. "; J;
   "____"; A$(J-1)
170: NEXT J
180: END

```

入力ループ

比較・並べ替えループ

並べ替え

出力ループ  
( ☐ DEF ☐ A と押せばこのループのみ実行)

上記プログラムを実行してください。データは次の通りです。

HIROSI	AKIKO
SATUKI	AKIO
AKIKO	HIROSI
KAZUMI	KAZUMI
AKIO	SATUKI

プログラムを実行し、左の名前を入れれば右のように並べ替えられます。

## = 式の中での文字列の長さについて =

文字列の結合(加算)や、大小比較などの式において、文字列の長さ(文字数)の合計は80文字以内でなければなりません。80文字を超えますと文字記憶エリア(計算や比較のため、文字を一時記憶しておくところ)の容量をオーバーしてエラーになります。(ERROR 15)




たとえば、次の式では文字列の長さの合計は9文字になります。




X=LEN("ABC")+LEFT\$("DEFGHI",2))

## 11. リザーブ機能

**Q** **W** …… **P** キーには、それぞれにBASICの命令が割り当てられていて、各命令を簡単に入力することができるようになっていますが、ほかにBASICの命令や、よく使う計算式などを自由に割り当てることのできるキーがあります。


キーボードには、キー上になにも書かれていないキーが6個ありますが、このキー（ソフトウェアキー）に必要な命令や計算式などを割り当て、記憶させておく（リザーブする）ことができ、その内容を利用することができます。

たとえば    …キーにそれぞれSIN、COS、TAN、……などをリザーブしておけば、SIN30+COS30を入力するときは

 30   30  
(SIN) (COS)

と操作するだけで済みます。

なお、各キーにリザーブした内容を示すため、リザーブした内容に合わせて、それらを示す文字や記号など（キーシンボル）を計算機に記憶させ、必要に応じて表示部に表示させることもできます。

ソフトキーは6キー有り、1キーごとにリザーブしますが、リザーブしたい内容が7種類以上あるときは6キーでは足りません。このような場合には、表示部の右上に表示されているリザーブナンバーを  で切替えれば、それぞれのキーに重ねてリザーブすることができます。

リザーブナンバーはⅠ、Ⅱ、Ⅲと3つあります。したがって1つのキーには3種類の内容までリザーブすることができます。

### 11-1) リザーブのしかた


#### ◎ 準備








- ① RESERVEモードを指定します。 **SHIFT** **MODE** と押して表示部にRESERVEシンボルを表示させてください。
- ② すべての内容を新しくリザーブするときは、NEW **ENTER** と押し、前にリザーブされていた内容をすべて消してください。ただし、前の内容を残しておきたいときはこの操作は不要です。


●リザーブ内容をすべて消したときは、 **SHIFT** キーを押さなくてもソフトウェアキーのみで Ⅰ、Ⅱ、Ⅲ…などの記号を入力することができます。

#### ◎ リザーブ

次の手順でリザーブします。

- ①  キーを押して、表示部右上にⅠ、Ⅱ、Ⅲのいずれかのリザーブナンバーを表示させてください。（たとえばⅠを表示させてください。）
- ② そののち、次の形でリザーブしてください。

 リザーブしたい内容を入力 **ENTER**  
↑  
ソフトキー(       )

- ③ リザーブする内容が多くあり、1つのリザーブナンバーの中で、各キーにリザーブし切れないときは  を押し、ナンバー表示を替えて、リザーブしてください。

(例)

		(RESERVEシンボルを表示させてください。)	
(NEW		)	
		(リザーブナンバーⅠを表示させてください。)	
	SIN		F 1 : SIN
	COS		F 2 : COS
	A * A + B * B		F 3 : A * A + B * B
		(リザーブナンバーⅡを表示させてください。)	
	RUN		F 6 : RUN
	WAIT		F 5 : WAIT
	FOR		F 1 : FOR
	TO		F 2 : TO
	STEP		F 3 : STEP
	NEXT		F 4 : NEXT

リザーブナンバーⅠ  
でのリザーブ

リザーブナンバーⅡ  
でのリザーブ

○ソフトキーには左から、それぞれF 1、F 2、…F 6のキーナンバーがつけられています。キーナンバーは、今どのキーが指定されているか、あるいはリザーブ内容がどのキーにリザーブされているものかを示します。

○本機はこのリザーブのために、110バイトの容量が用意されています。リザーブはキーナンバーも含めて、110バイト以内でおこなってください。  
次にそれぞれの文字や命令などのバイト数の例を示します。

- ① キーナンバー……………1バイト


(例) F 1 :、F 2 :……………

- ② 数字・符号……………1バイト

(例) +、-、1~0……………E、√、π

- ③ 関数・ベーシック命令……………2バイト

(例) SIN、COS、SQR、PI、INPUT、RUN、IF……

○1つのキーにリザーブできる内容の長さはキーナンバーや  キーを含めて最大80キー命令です。


- キー命令はおおむね表示1桁に対し、1キー命令となります。(ソフトキーはキーナンバーに表示3桁を使用している通り、3キー命令になります。)


(例)  SIN A + COS B  F 1 : SIN A + COS B

13キー命令 8バイト




## ◎ リザーブ内容の訂正・変更

リザーブ内容の訂正や変更は、マニュアル操作で説明しましたキー操作の訂正やプログラムを修正するときと同様の方法でおこないます。

(例) リザーブナンバーⅠ、キーナンバーF2(  キー)にリザーブされているCOSをACSに変更したいとき

 を操作してⅠシンボルを表示させます。

 F2: COS


 または  を操作 F2:  COS

↑ ここを点滅させます。

ACS  F2: ACS


## ◎ リザーブ内容を消去したいとき

○リザーブした内容をすべて消去したいときは


RESERVEモードで NEW 


と押してください。

○1つのキーにリザーブされている内容のみを消去することはできません。

このようなときは、各キーが持っているキャラクターシンボル(!, ", #, ... &)をリザーブしてください。(NEW  で消去された後、なにもリザーブされていないキーは、すべてこれらのキャラクターシンボルが呼び出されます。)

次にその操作例を示します。



(例) リザーブナンバーⅠ、キーナンバーF3(  キー)にリザーブされているA\*A+B\*Bの替りに#記号をリザーブします。

 を操作してⅠシンボルを表示させます。

 を操作 F3: A\*A+B\*B

 または  を操作 F3:  A\*A+B\*B

  F3: #  A+B\*B


  を操作してあとの内容をすべて消します F3: #\_

 F3: #

## ◎ リザーブした内容の利用のしかた

リザーブした内容はRUNモードやPROモードで、計算式やプログラムを書き込むときなどに利用します。

(例) 先に、ソフトキーにリザーブした内容を用いてプログラムを書き込んでみましょう。

 でⅠシンボルを表示させてください。

PROモードにしてください。

10 FOR A=0 <sup>I</sup>  を押す TO 90 <sup>II</sup>  を押す STEP 5 <sup>III</sup>  を押す

20 B=SIN A

<sup>IV</sup>  を押す  
30 WAIT 60

40 PRINT "SIN ": A: "= ": B

<sup>V</sup>  を押す  
50 NEXT A

60 END

このようにキーの操作が、かなり少なくなります。

次にRUNモードにして、いま書き込んだプログラムを実行してみましょう。

ここでも  キーにリザーブされているRUN命令を用いれば

**ENTER**

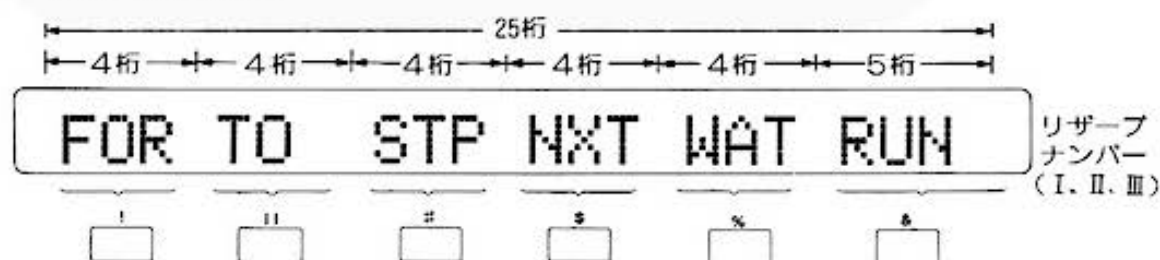
と押すだけで実行を開始します。

なお、たとえばRUN@のようにリザーブしておけば@が **ENTER** キーと同じに働きますので、ソフトウェアキーだけで実行を開始させることもできます。

## 11-2) ソフトウェアキーのキーシンボル表示

ソフトウェアキーにリザーブした内容がすぐにわかるようにするため、それぞれのキーの位置に、リザーブした内容を表わす文字や記号をキーシンボルとして表示させるようにします。(下図参照)

### ◎ リザーブキーと表示位置



このような形で文字や記号を表示させます。

### ◎ キーシンボルの書き込みかた

キーシンボルは、表示される内容すべてを1つの文字列として書き込みます。したがって、先の表示の例では、次のようにFORからRUNまですべてを" "で囲って書き込みます。(" "内はスペースも含めて25字以内にしてください。)



(例) **SHIFT** **MODE** と押してRESERVEシンボルを表示させます。

**⇓** キーを押してリザーブナンバーⅡを表示させます。

"\_FOR\_TO\_STP\_NXT\_WAT\_RUN" **ENTER**

このとき、リザーブした内容のリザーブナンバーと、キーシンボルを書き込むときのリザーブナンバーを一致させておかなければ、キーシンボルと、リザーブされている内容が異ってしまいますので、必ずリザーブナンバーを一致させてください。

ソフトキーには3種類の内容がリザーブできますので、それに合わせて、キーシンボルも3種類書き込んでください。先に述べましたように、リザーブナンバー(Ⅰ、Ⅱ、Ⅲ)を一致させることを忘れないでください。

(例) 前の例に続いてナンバーⅠのキーシンボルを書き込みます。

**⇓** **⇓** と押してリザーブナンバーⅠを表示させます。

"\_SIN\_ACS" **ENTER**

○キーシンボルは、あなたにとってわかりやすければ、どのような文字、記号を用いてもかまいません。キーシンボルは、単にリザーブされている内容を読み出すときの目安となるものです。

○キーシンボルを新たに書き込めば、前に書かれていたシンボルは消されます。


注) キーシンボルを書き込むとき、" "内は必ず25字以内(スペースを含む)にしてください。" "内が25字を超えると正しく書き込まれないことがあります。


## ◎ キーシンボルの利用のしかた


(例) 先にリザーブナンバーⅡにリザーブした内容とキーシンボルを利用して、次のプログラムラインを書き込みます。

```
10 FOR A=0 TO 90 STEP 5
20 B=SIN A
```


## PROモード

↑ ナンバーⅡを表示させます。(キーシンボルが表示されます。)  
 FOR TO STP NXT WAY RUN

10  A=0 10 FOR A=0 \_



↑ キーシンボルを呼び出してキーを探します。  
 FOR TO STP NXT WAY RUN


Ⅱ  90 10 FOR A=0 TO 90 \_


↑ キーシンボルを呼び出してキーを探します。  
 FOR TO STP NXT WAT RUN


Ⅴ  5  10: FOR A=0 TO 90 STEP 5

20 B= 20 B= \_




↑ ナンバーⅠを表示させます。(キーシンボルが表示されます。)  
  SIN ACS

Ⅰ  20 B=SIN \_

A  20: B=SIN A

-  キーはキーシンボル表示と、現在表示されている内容を入れ替えます。
- キーシンボルの呼び出しは、必要な内容がリザーブされているキーを探すためですので、既にわかっている場合は必要ありません。

## ＝参 考＝

ソフトキーを定義付けキーと同じように使用して、プログラムの実行を開始させることができます。たとえば  に GOTO 100@ とリザーブしておけば、RUNモードで  を押すだけで100ラインへジャンプして、プログラムの実行を開始します。(@が  キーの働きをします。)

## カラーグラフィックプリンタについて

この計算機用として別売のカラーグラフィックプリンタ(CE-150)(カセットインターフェイス内蔵)が用意されています。CE-150をお使いいただくことにより、計算結果やプログラムを書き出したり、図形を書いたりすることができます。また、CE-150をお使いいただくことにより、テープレコーダーを外部記憶装置として使用することができ、カセットテープなどにプログラムやデータなどを記憶したり、その内容をテープから読み込んだりすることができます。テープレコーダーは2台接続することができますので、一方を読み込み用、もう一方を記録用として、ばらばらに記録されているプログラムを一本のテープにまとめたり、一方からデータを読み込んで処理し、もう一方に記録していくなど、応用範囲も広がります。

### 〈CE-150と計算機の接続〉

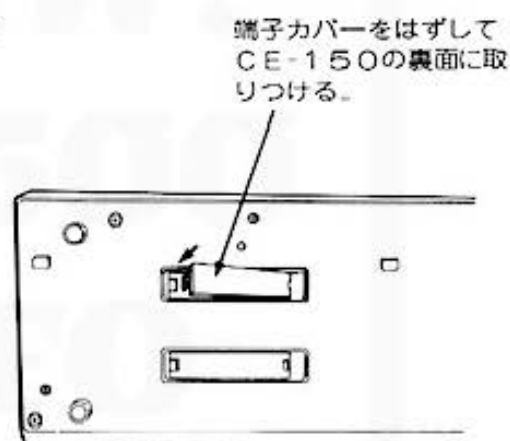
カラーグラフィックプリンタ CE-150と計算機 PC-1500を接続するときは、次の手順でおこなってください。

- (1) **OFF** キーを押して計算機の電源をオフにしてください。
- (2) 計算機の左側面にある端子カバーをはずし、プリンタの裏面の保存場所に取りつけてください。

図1

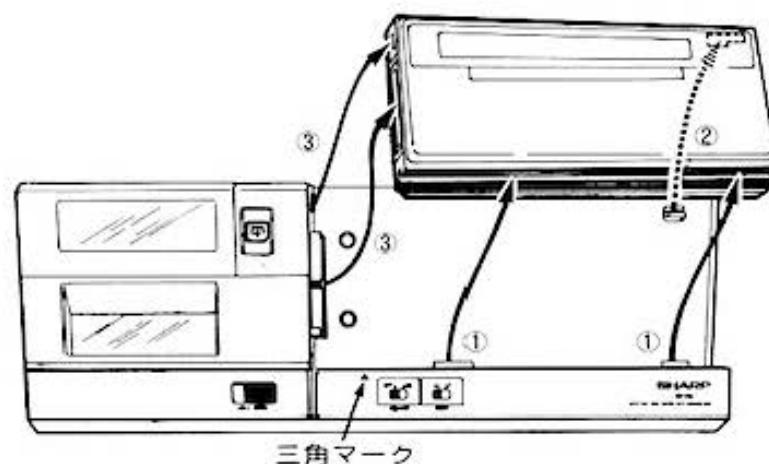


図2



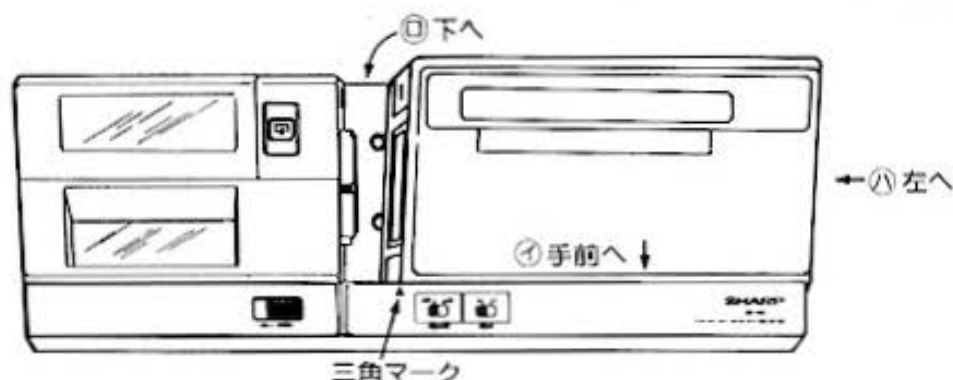
- (3) 図の矢印のようにプリンタの突起部分を計算機のミソ部分に入れていきますが、入れていく順序は矢印の番号および次の説明順序通りです。

図3



- ① ▲の部分に計算機の左側面がくるように計算機をプリンタに乗せてください。

図4



- ②図3の矢印②の部分が入らないときは、計算機を右側、あるいは左側へ軽く動かしてください。

- ③矢印②の部分が入ったら図3の矢印③の部分(計算機とプリンタ)を差し込んでください。

●接続端子は確実に差し込んでください。ただし、むりやり差し込まないでください。

注) 計算機をプリンタに取りつけるとき、あるいは取りはずすときは **OFF** キーにより計算機の電源をオフにしておこなってください。

電源をオンのままで取りつけたり、取りはずしたりしたときは計算機のすべてのキーが動かなくなることがあります。万一このような状態となったときは **ON** キーを押しながら計算機裏面のオールリセットスイッチを押してリセットしてください。このとき計算機の内容はすべてクリアされますのではじめから操作してください。

注) 計算機とプリンタが接続されている場合に、 **ON** キーにより計算機の電源を入れたとき、

NEW0? : CHECK 6

あるいは

: CHECK 6

のようにチェックメッセージが表示されることがあります。また、使用中に ERROR 80 とエラーメッセージが表示されることがあります。

これらは、プリンタの電池の電圧低下などにより、プリンタが正常に動作できないことを示しています。このような場合は、CE-150の取扱説明書をご覧のうえ、充電をおこなってください。

なお、上記メッセージが表示された場合はプリンタがロックされます。これを解除するには充電をおこなったのち、 **OFF** **ON** と押してください。

## プリンタに関する命令

これ以降に説明しますプリンタに関する命令およびテープレコーダーに関する命令(96ページ)は、別売のカラーグラフィックプリンタ(カセットインターフェイス内蔵)CE-150が持っている命令で、計算機本体にはありませんので、CE-150を接続しているときにのみ命令としてプログラミングすることができます。したがって、これらの命令を用いてプログラミングするときは、必ずCE-150を接続してください。

(注) プリンタのペン交換の方法や、ペーパーのつけかたなどについてはCE-150の取扱説明書をご覧ください。

### (1) モード指定命令(TEXT, GRAPH)

本プリンタは、計算結果や、プログラムリストなどの文字や数字を印字する場合と、グラフなど図形を書く場合のように、その使用目的により2つのモードがあります。

#### ◎ TEXT命令(テキスト)

文字や数字を印字するモード(TEXTモード)を指定する命令で、一般形は次のようになります。

TEXT  
(TEXT ENTER)

なお、OFF キーで切られた電源を ON キーで入れたときはTEXTモードが指定されます。また、後で述べますLIST命令、TEST命令を実行した場合やマニュアル操作で計算結果などを印字した場合は自動的にTEXTモードが指定されます。

#### ◎ GRAPH命令(グラフィック)

グラフなどの図形を書くモード(GRAPH)を指定する命令で、一般形は次のようになります。

GRAPH  
(GRAPH ENTER)

このモードが指定されると、ペンを左端にもどして、その位置をこれから図形などを書く場合の原点にします。

### (2) プリント命令(LPRINT)

#### ◎ LPRINT命令(ラインプリント)

計算機の表示命令として、PRINT命令がありましたが、これに替わるプリンタの印字命令としてLPRINT命令があります。

このLPRINT命令の一般形はPRINT命令とほとんど同じ形で指定することができます。

① LPRINT  $\left\{ \begin{array}{c} \text{式} \\ \text{文字変数} \\ \text{"文字"} \end{array} \right\}$

○TEXTモードにおいて、ペンが左端にあるときは、数値を右づめで、文字を左づめで印字します。もし、ペンが左端にないときは、文字・数値ともそのペン位置から印字されます。

②  $LPRINT \left\{ \begin{array}{c} \text{式} \\ \text{文字変数} \\ \text{"文字"} \end{array} \right\}, \left\{ \begin{array}{c} \text{式} \\ \text{文字変数} \\ \text{"文字"} \end{array} \right\}$

○TEXTモードにおいて、文字サイズ1(1行36桁)のときは左右2分割して、指定された内容を印字します。ただし、数値が1行に印字可能な桁数を超えた場合はエラーになります(ERROR 76)

文字が1行に印字可能な桁数を超えた場合は、超えた部分は無視されます。

(1行に印字できる桁数については下の表を参照ください。)

③  $LPRINT \left\{ \begin{array}{c} \text{式} \\ \text{文字変数} \\ \text{"文字"} \end{array} \right\};$

○TEXTモードにおいて、現在のペン位置から印字をおこないます。印字内容が1行に印字できないときは改行して印字されます。

### (3) 文字指定命令(CSIZE, ROTATE, COLOR)

文字の大きさや文字の向き、また何色で印字するかを指定する方法を説明します。

#### ◎ CSIZE命令(キャラクタサイズ)

文字の大きさを指定する命令で、次のような形で指定します。

CSIZE 式

式の値を1～9にすることにより印字する文字の大きさを指定します。

それぞれの値に対する文字の大きさ、横1行に印字できる桁数は次の表のようになります。

CSIZE	1	2	3	4	5	6	7	8	9
桁数	36	18	12	9	7	6	5	4	4
文字高さ(mm)	1.2	2.4	3.6	4.8	6.0	7.2	8.4	9.6	10.8
文字幅(mm)	0.8	1.6	2.4	3.2	4.0	4.8	5.6	6.4	7.2

●TEXT命令およびGRAPH命令が実行されますと、文字サイズは自動的に2が指定されます。

●CSIZE 1(もっとも小さい文字)が読み取りにくい場合は CSIZE 2 をご使用ください。

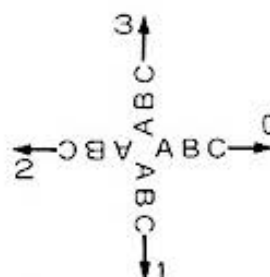
#### ◎ ROTATE命令(ローテイト)

ROTATE命令は文字の向き(印字方向)を指定する命令で、次のような形で指定します。

(GRAPHモードでのみ有効な命令です。)

ROTATE 式

式の値を0～3にすることにより、文字方向が右に示すようになり、矢印方向に向かって印字がおこなわれます。



●TEXT命令およびGRAPH命令が実行されますと、文字の向きは自動的に 0 の方向が指定されます。



◎ **COLOR命令(カラー)**

COLOR命令は印字する文字や線の色を指定する命令で、次のような形で指定します。

COLOR 式

式の値を0～3にすることにより、文字などの色を指定します。文字の色は黒、青、緑、赤の4色あり、それぞれの色を指定します。

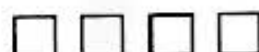
- プリンタには4本のプリンタ用ボールペンを取りつけます。このボールペンの色を黒、青、緑、赤にすれば、COLOR命令により、指定された位置に取りつけられている色で印字をおこなうことができます。なお、電源を入れたときはCOLOR 0に相当する位置に取りつけられている色(ボールペン)が選ばれます。

(4) **色やインキのチェック(TEST)**◎ **TEST(テスト)**

色の指定のチェック(COLOR命令で0～3に指定される色のチェック)やインキ量、インキの乾燥をチェックするための命令で、次のような形で指定すれば、5mm×5mmの正方形をそれぞれの色で書きます。

TEST

(TEST ENTER)



0 位置 の色	1 位置 の色	2 位置 の色	3 位置 の色
---------------	---------------	---------------	---------------

(5) **印字位置の指定(TAB)**◎ **TAB命令(タブ)**

計算機のCURSOR命令が、表示の桁単位で表示開始位置を指定するのと同様、TAB命令はペンの位置を指定のポイントまで移す命令で、次のような形で指定します。(TEXTモードでのみ有効)

① TAB 式

② LPRINT TAB 式1; 式2; ……

TABに続く(式1)の値によりペンの位置を指定します。ただし、文字の大きさにより、ペンの移動量が異なりますので、式の値の範囲も次の表に示す通り、変化します。

文字の大きさ指定	CSIZE 1	CSIZE 2	CSIZE 3	CSIZE 4	CSIZE 5	CSIZE 6	CSIZE 7	CSIZE 8	CSIZE 9
式の範囲	0～35	0～17	0～11	0～8	0～7*	0～5	0～5*	0～4*	0～3

※CSIZE 5、7、8において、それぞれTAB7、TAB5、TAB4を指定した場合は続いて印字をおこなうとき、改行あるいはエラーになります。

○次のように指定した場合は1行改行される命令になります。

LPRINT TAB 式; \_\_\_\_\_ 1行改行を指定  
 ↑ 印字する内容を指定しません。

(例) 文字Aを用いてサインカーブを書いてみましょう。

```
10:TEXT :CSIZE 1
20:FOR H=0 TO 360
    STEP 8
30:A=SIN H
40:B=INT (A*17)
50:TAB (B+18)
60:LPRINT "A"
70:NEXT H
80:END
```

RUN



## (6) 紙送り命令(LF)

### ◎ LF命令(ラインフィード)

LF命令は紙送りをおこなう命令で、次の形で指定します。(TEXTモードでのみ有効)

LF 式

式の値で示す行数の紙送りをおこないます。(式の値は32767までです。)

式の値が負のときは逆方向へ最大10.24 cmまで紙送りをおこないます。

**(7) ペン移動命令 (GLCURSOR)****◎ GLCURSOR命令(グラフィックラインカーソル)**

ペンの位置を原点からX軸、Y軸方向に移動させる命令で、次のような形で指定します。

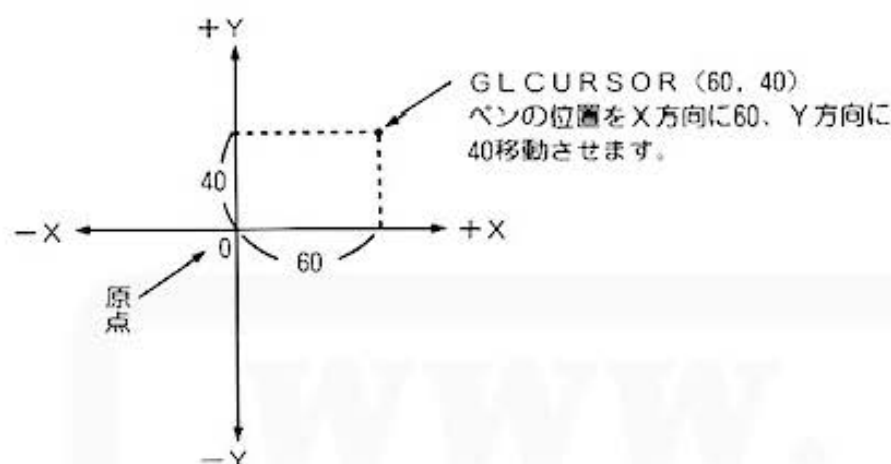
(GRAPHモードでのみ有効)

GLCURSOR (式1, 式2)

式1の値でX軸方向へ、式2の値でY軸方向への移動量を指定します。

式の値は-2048~+2047まで指定可能で、ペンの最小移動量は0.2mmになります。

(例)



注) +Y方向にペンを移動させることはペーパー(紙)を逆方向に送ることになります。この場合は最大10.24 cmまでしか送られませんので注意してください。

**(8) 原点指定命令 (SORGN)****◎ SORGN(セットオリジン)**

現在あるペンの位置を新しい原点に指定する命令で、一般形は次のようになります。

(GRAPHモードでのみ有効)

SORGN

GRAPHモードが指定されたときはペンが左端にもどり、その位置が原点になります。図形を書く場合などでは、原点が左端にあると書きにくい場合などがありますので、たとえばGLCURSOR命令で、任意の位置にペンを移動させ、SORGNで原点指定をして、その位置を基点として図形などを書きます。

## (9) 描線命令(LINE, RLIN)

指定位置から指定位置まで線を書く命令です。

### ◎ LINE命令(ライン)

次の形で線を書く位置、線の種類、色などを指定します。(GRAPHモードでのみ有効)

〈一般形〉

LINE(式1, 式2)-(式3, 式4), 式5, 式6

(A)

(B)

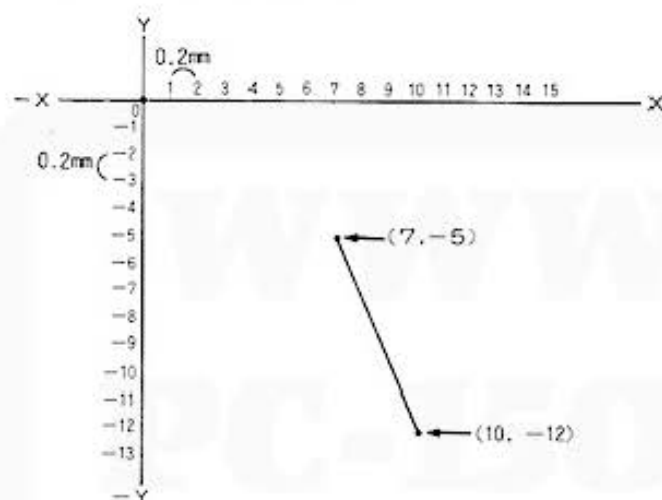
(C)

(D)

○線の位置、長さ、方向の指定

(式1, 式2)で示される座標位置から、(式3, 式4)で示される座標位置まで線を書きます。

なお、(式1, 式2)を省略することができます。この場合はペン位置から(式3, 式4)で示される位置まで線を書きます。



○線の種類

式5の値によりそれぞれ次のように線の種類を指定します。

0 : 実線

1 ~ 8 : 破線

9 : 線を書かずにペンを移動

(例)

- 1 ..... (dotted line)
- 2 ..... (dashed line)
- 3 ..... (dash-dot line)
- 4 ..... (long dashed line)
- 5 ..... (short dashed line)
- 6 ..... (medium dashed line)
- 7 ..... (long dash short dash line)
- 8 ..... (short dash long dash line)

## ○線の色

式5の値により線の色を指定します。式5の値は0～3で指定します。

- 線の色については86ページのCOLOR命令を参照してください。

なお、式5、式6は省略することができます。式5や式6が省略されたときは、それ以前に指定されている値(状態)で実行します。

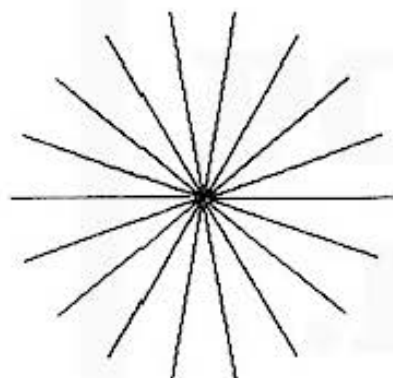
(例)

```

10:GRAPH:RANDOM
20:GLCURSOR(107, ←図形を書くため、ペン位置をペーパーの中ほどに
  -120):SORGN      ←移動させて、その位置を原点に指定します。
30:FOR J=0TO 340
  STEP 20
40:A=107*COS J
50:B=107*SIN J
60:R=RND 4-1        ←0～3の乱数をRに代入
70:LINE(0,0)-(A,    ←Rの値により色を指定します。
  B),0,R
80:NEXT J
90:END

```

RUN



## ○四角形を書くには

LINE命令の最後にBを入れることにより、(式1, 式2)で示される座標位置と(式3, 式4)で示される座標位置を対角線とする四角形を書きます。ただし、式1と式3あるいは式2と式4が同じ値の場合は四角形にならず、直線が書かれます。

(例) LINE(10,50)-(200,-20),2,0,B



○連続して線を書く場合の指定のしかた

次のように「LINE」命令を書くことにより、連続した線をつぎつぎに書くことができます。

$$1 \rightarrow NE(\mathfrak{F}_1, \mathfrak{F}_2) \rightarrow (\mathfrak{F}_3, \mathfrak{F}_4) \rightarrow (\mathfrak{F}_5, \mathfrak{F}_6) \rightarrow \cdots \rightarrow (\mathfrak{F}_{11}, \mathfrak{F}_{12}), \mathfrak{F}, \mathfrak{F}$$

このように指定すれば(式1、式2)の点から(式3、式4)の点まで線を書き、続いて(式3、式4)の点から(式5、式6)の点まで線を書く……というように連続して線を書くことができます。なお、(式、式)は6つまで指定することができます。

◎ RLINE命令(レティブライン)

この命令の一般形は、I I NF命令と同じ形になります。

(一般形)

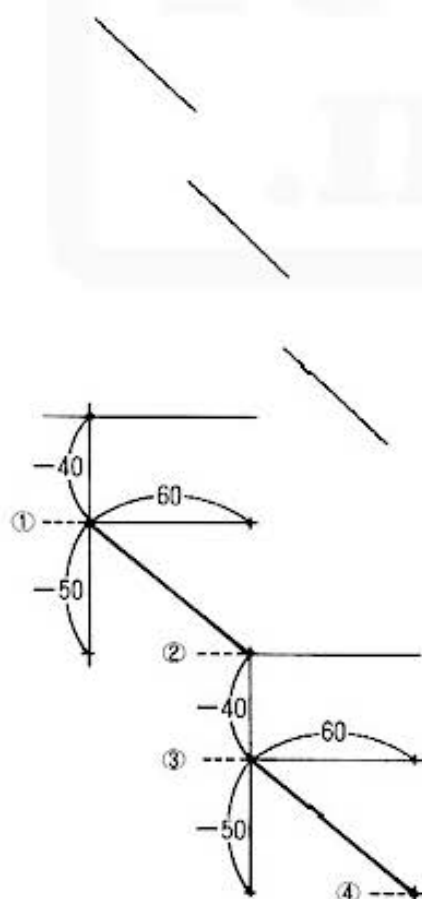
$$\text{RLINE}(\text{式1, 式2})-(\text{式3, 式4}), \text{式5, 式6}$$

(A)
(B)

LINE命令とRLINE命令との違いは、LINE命令が、SORGN命令などにより指定された原点を基準にし、その原点からの座標により各点の位置を指定するのに対し、RLINE命令は、ペン位置を原点にしてその位置からの座標により次の点の位置を指定します。そのほかはLINE命令と同様の形で使用します。

```
(例) 10:GRAPH
      20:FOR A=1TO 3
      30:RLINE (0,-40)-
          (60,-50)
      40:NEXT A
      50:END
```

このプログラムを実行すれば次のようにプリントされます。



①の位置を原点として②の位置を決め、  
②の位置を原点として③の位置を決めます。  
そして③の位置を原点として④の位置を決  
めています。



## ⑩ リストのしかた(LLIST)

### ◎ LLIST命令(ラインリスト)

この命令は計算機に記憶されているプログラムを書き出す命令で、一般形は次のようになります。(PROあるいはRUNモードでのマニュアル操作により実行します。)

LLIST 式<sub>1</sub>, 式<sub>2</sub> **ENTER**

この形により、式<sub>1</sub>の値で示されるラインから、式<sub>2</sub>の値で示されるラインのプログラムまでが書き出されます。なお、式<sub>1</sub>、式<sub>2</sub>は省略することができ、これらを省略した場合はそれぞれ次のようになります。

LLIST **ENTER**

プログラムの先頭ラインから、すべてのプログラムを書き出します。

LLIST 式<sub>1</sub> **ENTER**

式<sub>1</sub>の値で示されるラインの内容のみを書き出します。

LLIST 式<sub>1</sub>, **ENTER**

式<sub>1</sub>の値で示されるラインから、最後のラインまでのプログラムを書き出します。

LLIST, 式<sub>2</sub> **ENTER**

プログラムの先頭ラインから、式<sub>2</sub>の値で示されるラインまでのプログラムを書き出します。

○リストされる文字の大きさはCSIZE 1 あるいはCSIZE 2 に限定されます。

CSIZE 2 以上の文字サイズを指定したときは、すべてCSIZE 2 で印字され、CSIZE 1 を指定したときはそのサイズで印字されます。

(注) MERGE命令(テーブルコーダーに関する命令： 98ページ参照)により、複数のプログラムが同時に計算機内に記憶されているときは、LLISTに続く式<sub>1</sub>、式<sub>2</sub>は最初のプログラムのラインを指定します。したがって、たとえば、LLIST 50,

**ENTER** と押した場合は、最初のプログラムの50ラインから最後のプログラムの最終ラインまでがリストアウトされます。

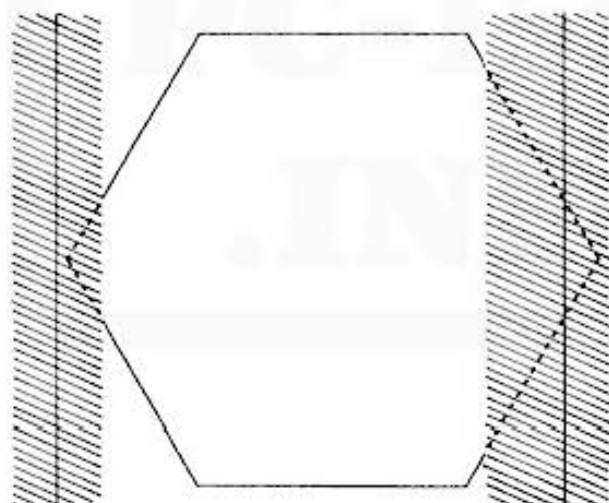
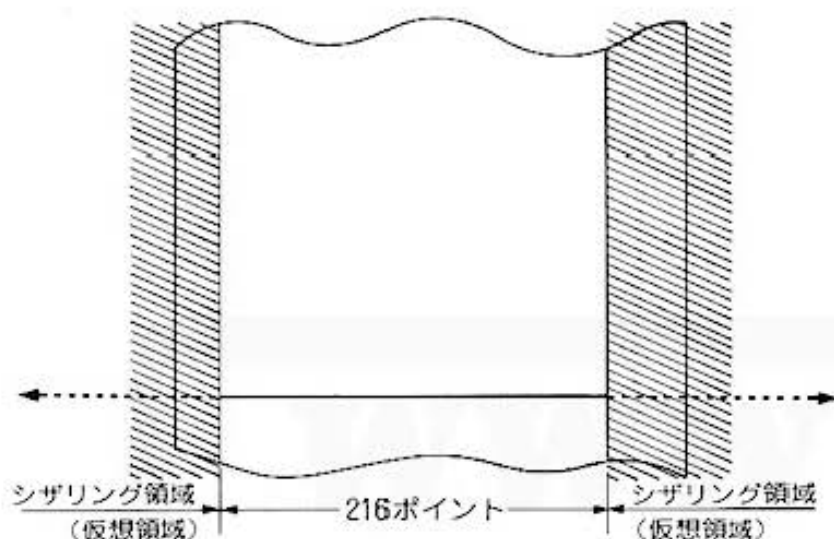
本プリンタで図形を書く場合は、できるだけ上下方向のくり返し(前後の紙送り)を少なくしたほうが、より正確な図形が描けます。

上下方向にくり返しが多い図形を書いたときは、紙送りの誤差が累積されて、図形のずれが大きくなることがあります。

したがって図形をプログラミングするときは上下方向のくり返しの少ないプログラムを用いてください。

## ＝ペンおよび紙の移動範囲について＝

本プリンタに使用する紙の幅は約58mmですので、図形を書く場合などはその大きさに制限があります。しかしながら、図形の一部のみを書き出したいが、全体を書き出すプログラムを作るほうが簡単になる場合や、全体を紙幅分ずつに分割してでも大きく書きたい場合などがあります。このようなときに便利な機能がシザリング機能です。この機能は、紙の上にあたる部分の図形は実際に書き、紙の上を外れる部分は、その部分にペンが移動できると仮定して、仮想的に図形を書く機能です。



```

10: GRAPH
20: GLCURSOR (130,
  -200): SORGN
30: C=150, D=0
40: FOR J=0 TO 360
  STEP 60
50: A=150*COS J
60: B=150*SIN J
70: LINE (C, D)-(A,
  B)
80: C=A, D=B
90: NEXT J
100: END

```

また、逆方向に紙を送った場合に10.24cmまでしか送られませんが、それ以上になる値をY軸方向に指定して図形を書く場合も、シザリング機能が働きます。

- シザリング機能は便利な機能ですが、プログラムを誤りますと仮想領域にすべての図形を書いてしまい、紙上にはなにも書かないことにもなりますので、プログラムを作るには十分注意が必要です。

## ＝ 逆紙送りカウンターについて ＝

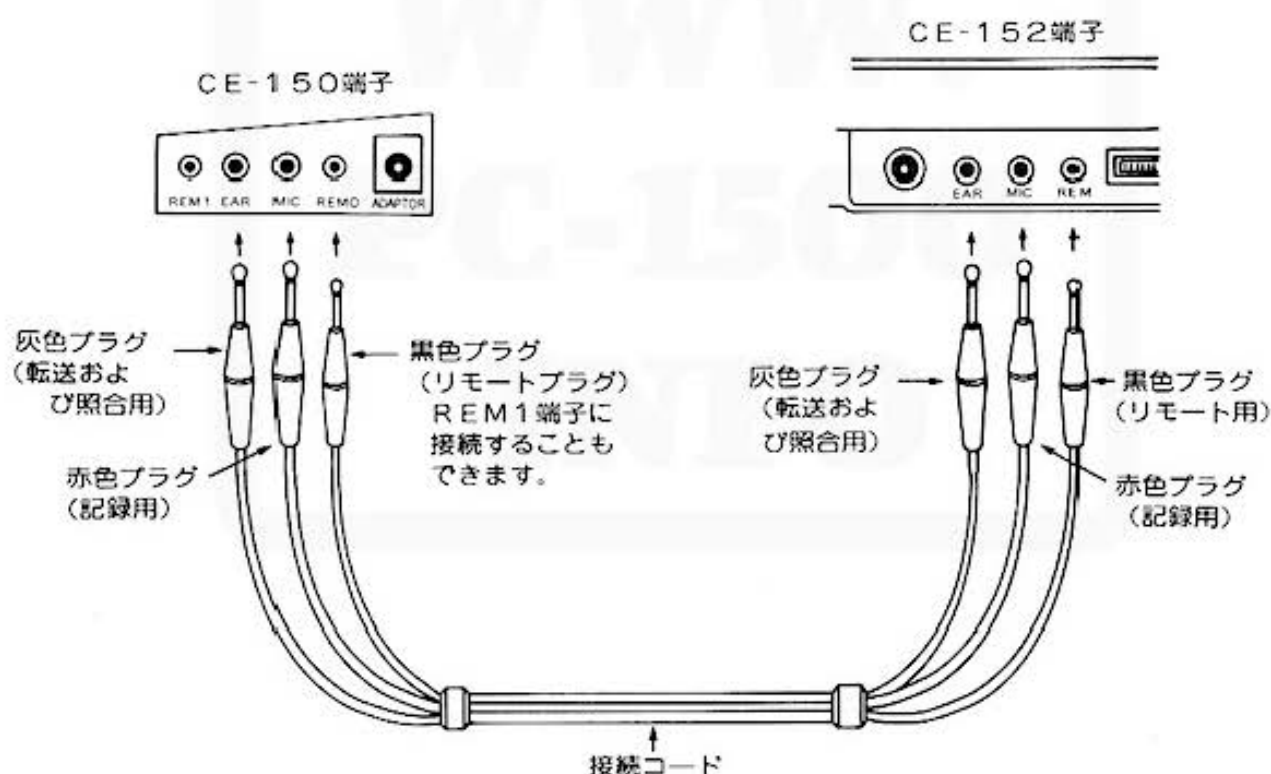
本プリンタには、逆紙送りにおいて、10.24cm以上紙を逆送りできないようになっています。この長さは常にカウンターによりカウントされていますので、図形を書くときの原点などとは関係なく、1番前まで送られた位置から逆方向に10.24cmまで送られる(もどされる)と、逆紙送りは止まります。(GRAPHモードではシザリング領域に入ります。)

## 〈テープレコーダーとの接続方法〉

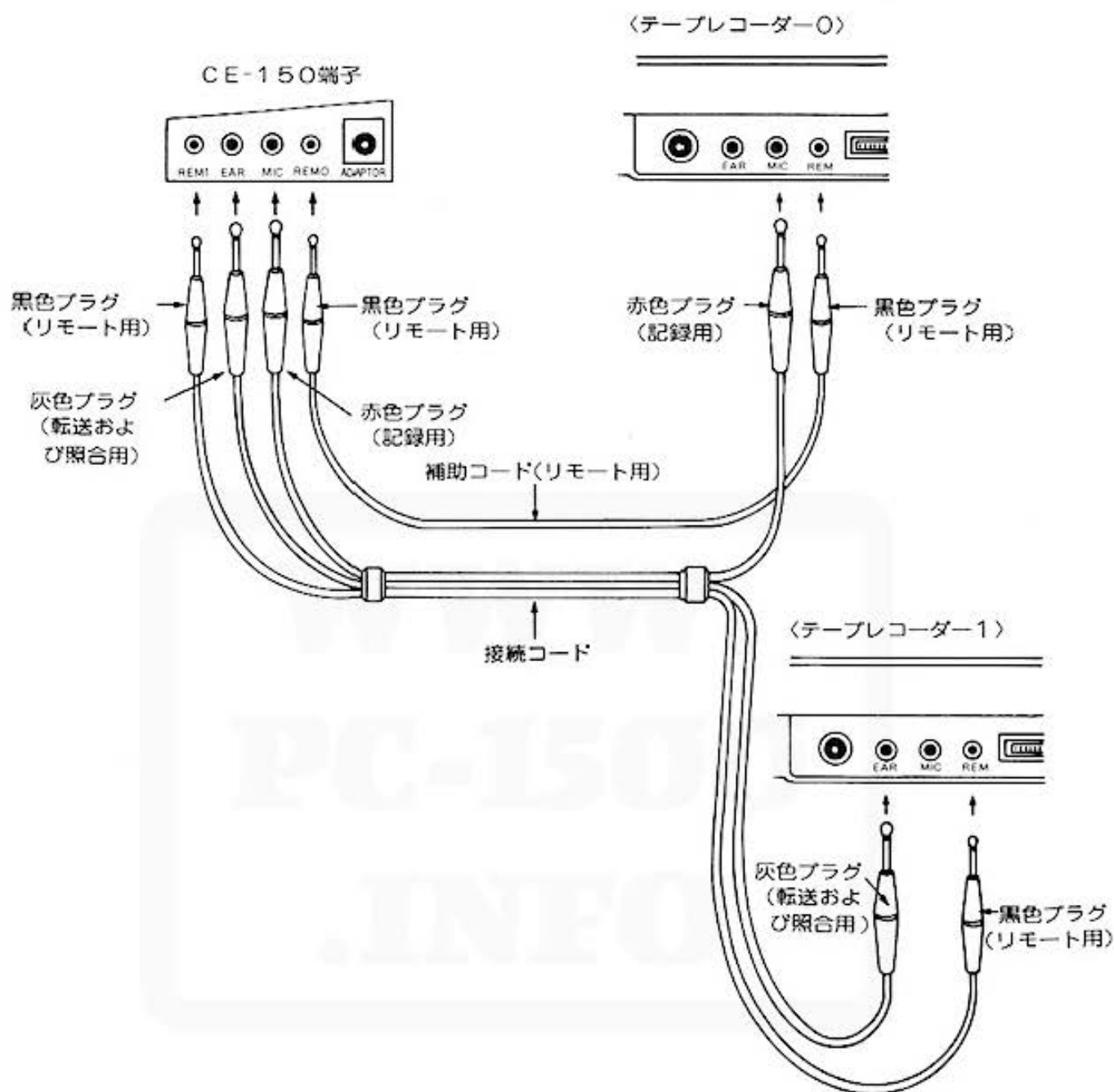
カラーグラフィックプリンタCE-150(カセットインターフェイス内蔵)にはテープレコーダーを1台あるいは2台接続して使用することができます。(テープレコーダーはシャープポケットコンピュータ専用カセットテープレコーダーCE-152をご使用ください。)

ここではCE-152との接続方法について説明します。

### ① テープレコーダー1台を接続する場合



② テープレコーダー2台を接続する場合



テープレコーダーを2台使用する場合は、どちらか一方を記録用とし、もう一方を読み込み用(転送用)とします。そして図のように接続コードと、リモート用補助コードとを用いて接続します。

- 計算機からプログラムあるいはマニュアル操作によりテープレコーダーの回転を止めたり、動かしたりさせるときにリモート端子を使いますが、CE-150にはREM 0とREM 1の2つのリモート端子があります。この端子はどちらを使ってもかまいませんが、プログラムなどでは、その命令の中でREM 0端子に接続されているテープレコーダーか、あるいはREM 1に接続されているテープレコーダーかを指定しますので、プログラムなどに合わせて接続する必要があります。

## テープレコーダーに関する命令

カセットテープなどにプログラムやデータを記録したり、計算機に読み込んだり（転送したり）、あるいは記録されている内容を照合したりする命令について説明します。なお、CE-150にはテープレコーダーを2台接続することができますので、説明の中でテープレコーダーを指定するとき、必要に応じて次のように記載します。

テープレコーダー0：CE-150のREM 0端子に接続しているテープレコーダー

テープレコーダー1：CE-150のREM 1端子に接続しているテープレコーダー

### (1) プログラムの記録命令(CSAVE)

#### ◎ CSAVE命令(カセット・セーブ)

プログラムやリザーブ内容をテープに記録する(セーブする)命令で、次のような形で指定します。

① CSAVE“ファイル名” **ENTER**

② CSAVE-1“ファイル名” **ENTER**

①の形はテープレコーダー0が指定され、②の形ではテープレコーダー1が指定されます。

#### ○プログラムの記録

プログラムをテープに記録する場合はPROモードかRUNモードでCSAVE(CSAVE-1)命令を実行します。

(例) PROモードかRUNモードを指定

CSAVE "PROG.-1" **ENTER**

ファイル名として「PROG.-1」をテープに記録したのち、計算機内のプログラムをすべて記録します。

#### ○リザーブ内容の記録

リザーブ内容をテープに記録する場合はRESERVEモードでCSAVE(CSAVE-1)命令を実行します。

(例) RESERVEモードを指定

CSAVE "RESRV-1" **ENTER**

ファイル名として「RESRV-1」をテープに記録したのち、リザーブ内容とキーシンボル内容をすべて記録します。

〔 記録をおこなったあと、プログラムなどが確実に記録されたことを確認するために、あとで説明しますCLOAD?命令を実行してください。 〕

- 注) ●同じテープの同じ面(A面あるいはB面)に、同じファイル名で内容の違うプログラム、あるいはリザーブ内容を記録することは避けてください。同じファイル名のものが2つ以上ありますと、テープから読み込む(転送する)ときにまちがった内容を読み込む恐れがあります。
- 1本のテープに複数のプログラムなどを記録するときは、プログラムとプログラムが重ならないように少し間をあけて記録してください。

記録をおこなうとき、テープレコーダーのテープカウンターの数字を紙などに控えておけば、転送などをおこなうとき、記録した部分を簡単に探すことができます。

#### ○ファイル名

ファイル名はプログラムなどにつける名前(見出し)のことで、ファイル名をつけてプログラムなどを記録しておけば、あとで転送などをおこなうとき、計算機によりファイル名を探し出して転送などをおこなうことができますので便利です。なお、ファイル名を省略することもできます。

## (2)プログラムの転送命令(CLOAD, MERGE, CHAIN)

転送するときはテープを巻きもどし、あるいは送って、転送する内容が記録されている部分の直前までテープを進めてください。

#### ◎ CLOAD命令(カセット・ロード)

テープに記録されているプログラムやリザーブ内容を計算機に転送するための命令で、次のような形で指定します。なお、この命令はマニュアル操作でのみ実行できます。

- ① CLOAD "ファイル名"       ( テープレコーダー0 )  
あるいは CLOAD       ( が指定されます。 )
- ② CLOAD-1 "ファイル名"       ( テープレコーダー1 )  
あるいは CLOAD-1       ( が指定されます。 )

#### ○プログラムの転送

テープに記録されているプログラムを計算機に転送する場合はPROモードかRUNモードでCLOAD命令を実行します。

(例) PROモードかRUNモードを指定

CLOAD "PROG.-1"

ファイル名が「PROG.-1」のプログラムを検索して、これを計算機のプログラムエリアに転送します。



ファイル名をつけずにCLOAD **ENTER** と操作した場合はテープレコーダーが動き始めて最初に記録されているプログラムを計算機に転送します。

#### ○リザーブ内容の転送

テープに記録されているリザーブ内容を計算機に転送させる場合はRESERVEモードでCLOAD命令を実行します。

(例) RESERVEモードを指定

CLOAD "RESRV-1" **ENTER**

ファイル名が「RESRV-1」のリザーブ内容を検索して、これを計算機のリザーブメモリに転送します。

ファイル名をつけずにCLOAD **ENTER** と操作した場合はテープレコーダーが動き始めて最初に記録されている内容を計算機に転送します。

注1) モードの指定をまちがえて転送をおこないますとエラーになります。

注2) 指定されたファイル名が検索できなかったときは、テープが回り終わっても計算機はファイル名を検索し続けます。この場合は **BREAK ON** キーを押して検索を止めてください。

これは、のちほど説明しますMERGE、CHAIN、CLOAD?、およびINPUT #命令でも同じです。

注3) CLOAD命令およびのちほど説明のCHAIN命令を実行中にエラーが発生しますと、計算機内のプログラムは無効になります。

#### ◎ MERGE命令(マージ)

MERGE命令はテープに記録されているプログラムを計算機に転送する命令で、CLOAD命令と同じ形で指定し、実行します。(マニュアル操作でのみ実行できます。)

① MERGE "ファイル名" **ENTER** (テープレコーダー0)  
あるいは MERGE **ENTER** が指定されます。

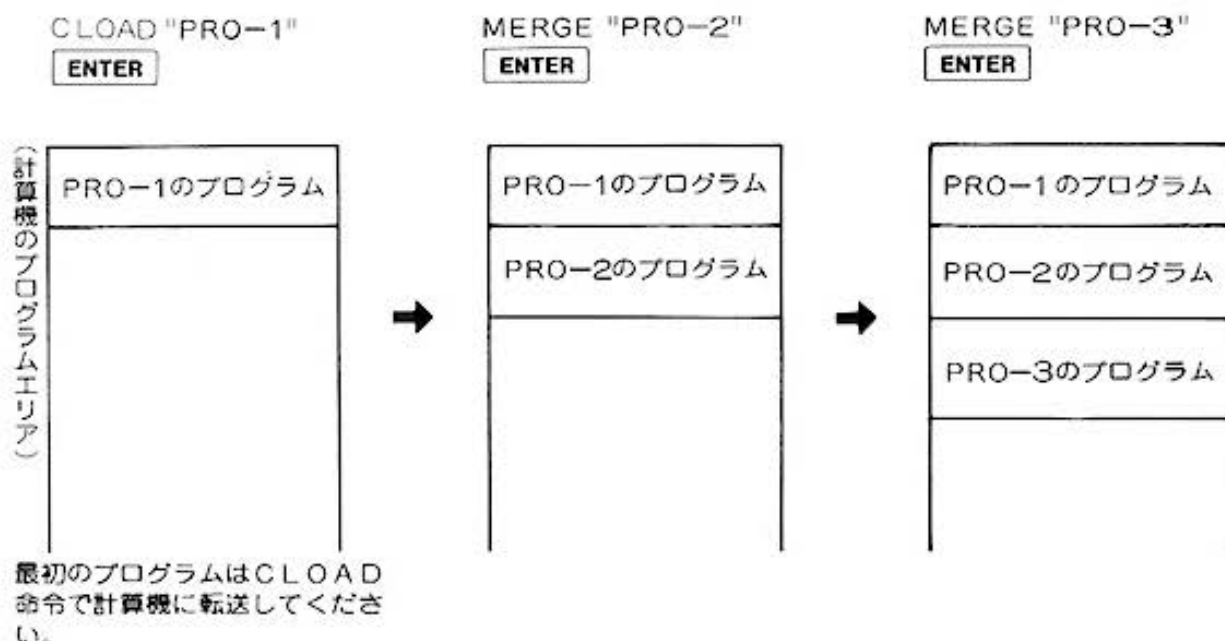
② MERGE-1 "ファイル名" **ENTER** (テープレコーダー1)  
あるいは MERGE-1 **ENTER** が指定されます。

CLOAD命令とMERGE命令の違いは、CLOAD命令が前に計算機に記憶されていたプログラムをすべて消してからテープに記録されているプログラムを読み込んでくるのに対して、MERGE命令は前のプログラムを保持したまま、それらの後に、テープに記録されている内容を追加していく形で読み込みます。したがって、全然別のプログラムを計算機に同時に記憶させておくことができます。

(例) テープにPRO-1～PRO-3のファイル名を持つプログラムが記録されているとき、PRO-1をCLOAD命令で、PRO-2、PRO-3をMERGE命令で計算機に転送した場合の記憶の状態は次のようになります。

(テープ)





MERGE命令により読み込まれたプログラムは上例のように記憶されますが、このとき、それぞれのプログラムに使用するラインナンバーやジャンプラベルなどに制約はなく、同じラインナンバーを持つプログラムを同時に記憶させておくことも可能です。

しかしながら、この場合2番目、3番目……に記憶されているプログラムを実行する方法がなくなりますので、これらのプログラムは事前に“定義付けプログラム”として、A、S ……などのキーに定義付けしておく必要があります。

(定義付けプログラムについては39ページを参照ください。)

なお、MERGE命令実行後のプログラム編集は最後に転送し、記憶されているプログラムに対してのみおこなうことができます。

#### ◎ CHAIN命令(チェーン)

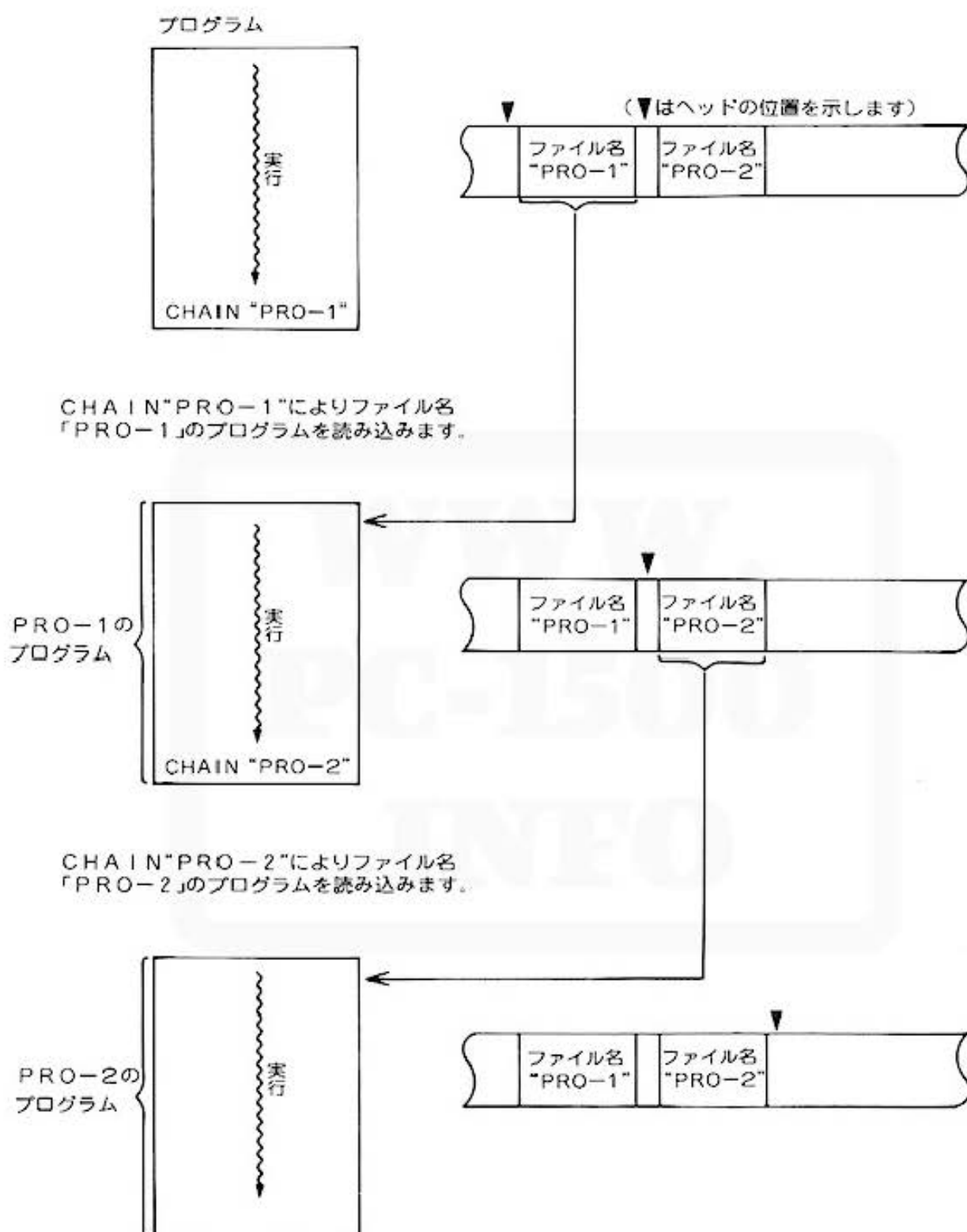
CHAIN命令は**プログラムの中で使用する命令**で、テープに記録されているプログラムを計算機に読み込んで、そのプログラムを実行していく命令です。(プログラムでのみ実行できます。) この命令を用いますと、計算機のプログラムエリア内に収まり切れないような長いプログラムでも、何分割かしてテープに記録しておき、次々と読み込みながら実行を続けていくことができます。

CHAIN命令の一般形は次のようになります。

- ① CHAIN “ファイル名”                      テープレコーダー0  
が指定されます。  
CHAIN “ファイル名”, 式
- ② CHAIN-1 “ファイル名”                      テープレコーダー1  
が指定されます。  
CHAIN-1 “ファイル名”, 式

指定されたファイル名のついたプログラムを読み込み、そのプログラムの先頭ライン、あるいは式の値で指定されたラインから実行します。なお、ファイル名を省略した場合は、テープが回り出して最初のプログラムを読み込みます。

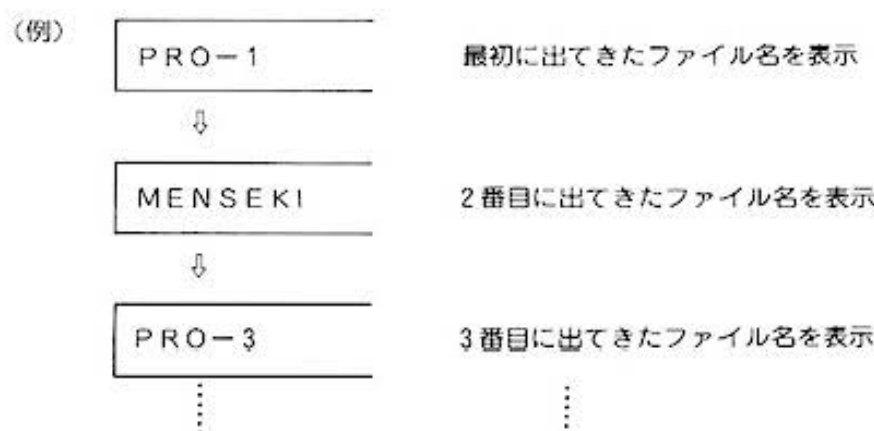
(例) CHAIN命令により、次のようにプログラムを読み込み、実行します。



このようにプログラムの最後にCHAIN文を入れておくと、それぞれのプログラムの実行の最後に、自動的に磁気テープから次のプログラムを読み込んで実行を続けていくことが可能になります。

## ＝ 検索中の表示 ＝

プログラムなどを検索しているとき、表示部には、途中でテープから読んだファイル名がづぎづぎと表示されます。



### (3) 照合命令(CLOAD?)

#### ◎ CLOAD? 命令(カセット・ロード?)

計算機に記憶させているプログラムあるいはリザーブ内容と、テープに記憶されている内容との照合をおこなう命令で、次のような形で指定します。なお、この命令は**マニュアル操作による実行のみ可能です**。

- ① CLOAD? "ファイル名" ENTER  
   あるいは CLOAD? ENTER
- ② CLOAD?-1 "ファイル名" ENTER  
   あるいは CLOAD?-1 ENTER

この命令により、もし不一致が生じたときは、エラー(ERROR 43)になります。不一致がなければプロンプト表示になります。

#### ○プログラムの照合

計算機に記憶されているプログラムとテープに記録されているプログラムを照合する場合はPROモードあるいはRUNモードでCLOAD?命令を実行します。

- (例) PROモードかRUNモードを指定  
 CLOAD? "PRO-1" ENTER

ファイル名が「PRO-1」のプログラムを検索し、その内容と、計算機のプログラムとを照合します。

#### ○リザーブ内容の照合

計算機に記憶されているリザーブ内容とテープに記録されているリザーブ内容を照合する場合はRESERVEモードでCLOAD?命令を実行します。

- (例) RESERVE モードを指定  
 CLOAD? "RESV-1" ENTER

ファイル名が「RESV-1」のリザーブ内容を検索し、その内容と、計算機のリザーブ内容とを照合します。

## (4) データの記録命令(P R I N T #)

### ◎ P R I N T #命令(プリント・クロスハッチ)

計算機内に記憶されているデータをテープに記録する命令で、次のような形で指定します。  
(マニュアル操作、プログラムとも実行できる命令です。)

- |   |   |   |
|---|---|---|
| ① | P R I N T #変数名, 変数名, ……<br>P R I N T #“ファイル名”; 変数名, 変数名, ……         | <div style="border-left: 1px solid black; border-right: 1px solid black; padding: 0 10px;">           テープレコーダー0<br/>が指定されます         </div>  |
| ② | P R I N T #-1, 変数名, 変数名, ……<br>P R I N T #-1, “ファイル名”; 変数名, 変数名, …… | <div style="border-left: 1px solid black; border-right: 1px solid black; padding: 0 10px;">           テープレコーダー1<br/>が指定されます。         </div> |

指定された変数に記憶されている内容をテープに記録します。なお、配列変数の場合は( )の中に\*を指定します。この形で、同じ名前の配列変数すべてを同時に指定することができます。配列変数(数値変数、文字変数とも)はこの形でのみ記録することが可能です。

(例) P R I N T #A, B, AA

変数A, B, AAの内容をファイル名なしでテープに記録します。

P R I N T # "DATA-1"; B(\*)

ファイル名「DATA-1」をつけて、B( )の変数(B(0), B(1), ……)をテープに記録します。

- (注) AAやB1\$のような2文字の変数はP R I N T #命令を実行する前に、プログラム・データエリアに確保されていなければなりません。  
もし確保されていない場合には計算機が、その変数名を探し続け(約30秒間)、その後、変数名が正しくないものと見なしてエラー(ERROR 21)になります。

## (5) データの転送命令(I N P U T #)

### ◎ I N P U T #命令(インプット・クロスハッチ)

テープに記録されているデータを指定された変数に転送する命令で、次のような形で指定します。(マニュアル操作、プログラムとも実行できる命令です。)

- |   |   |
|---|---|
| ① | I N P U T #変数名, 変数名, ……<br>I N P U T #“ファイル名”; 変数名, 変数名, ……         |
| ② | I N P U T #-1, 変数名, 変数名, ……<br>I N P U T #-1, “ファイル名”; 変数名, 変数名, …… |

テープに記録されているデータを指定された変数に転送します。なお、配列変数の場合は、INPUT #命令を実行する前にDIM命令で変数名と大きさを定義しなければなりません。

(例) INPUT # A, B, AA

テープが回りはじめて最初に記録されているデータを変数A, B, AAに転送していきます。

```
50 DIM B(5)
```

```
60 INPUT # "DATA-1" ;B(*)
```

ファイル名が「DATA-1」のデータを検索して、変数B(0)~B(5)に転送していきます。

(注) INPUT #命令で指定されている変数がAAやB1\$のような2文字の変数の場合、INPUT #命令を実行したときに、その変数がプログラム・データエリアに確保されていなければなりません。  
もし確保されていないときは、変数の形が正しくないものとしてエラー(ERROR43)になります。

したがって、このようなときは事前に代入文で変数を確保してください。

(例)    AA=0        適当な数値や文字などを代入することにより  
         B1\$="A"        変数を確保します。

INPUT # AA, B1\$   

注) INPUT #命令で指定した変数の数とテープに記録されているデータの数とが違っている場合は次のようになります。

- 指定した変数の数よりテープに記録されているデータの数が多い場合は、指定した変数にデータを転送し、残ったデータは無視します。
- 指定した変数の数よりテープに記録されているデータの数が多い場合は、記録されているデータが指定されている変数に順番に転送され、残った変数はそれまで記憶されていた内容を保持します。ただし、このとき計算機は次のDATAの入力待ち状態のままになっています。したがってこのときは  キーを押して動作を止める必要があります。
- INPUT #命令に続いて変数を指定していない場合はエラー(ERROR 43)になります。

## (6) リモートコントロール機能について (RMT OFF, RMT ON)

通常、CE-150のリモート端子(REM0、あるいはREM1)とテープレコーダーのリモート端子(REM)とを接続した場合、テープレコーダーの再生ボタン(▶)を押してもテープは回らず、計算機からの記録命令や転送命令などが実行されたとき、リモート機能が働いてテープが回ります。そして、記録や転送などが終了すればテープが止まります。(記録、転送、照合の名命令の-1がついた形のとき、CE-150のREM 1端子にコントロール信号が出され、-1がつかない形の命令ではREM 0端子にコントロール信号が出されます。)



## ◎ リモート機能の解除

プログラムの頭出し(プログラムの頭の部分を探し出すこと)をおこなうときや、記録されていない(空いている)部分を探すときなどはリモート機能を解除してからでないとテープが回りません。

○リモート機能の解除は次の方法でおこないます。

REM 0 端子：プリンタのリモートスイッチをOFF側にします。



テープレコーダー0はすべて手動で動作します。

REM 1 端子：RMT OFF 命令(リモート・オフ)を実行します。

RMT OFF **ENTER**

テープレコーダー1はすべて手動で動作します。

○解除したリモート機能をふたたびセットするには

REM 0 端子：プリンタのリモートスイッチをON側にします。



テープレコーダー0はリモート機能によりコントロールされます。

REM 1 端子：RMT ON 命令(リモート・オン)を実行します。

RMT ON **ENTER**

テープレコーダー1はリモート機能によりコントロールされます。

# 計算機とテープレコーダーの操作方法

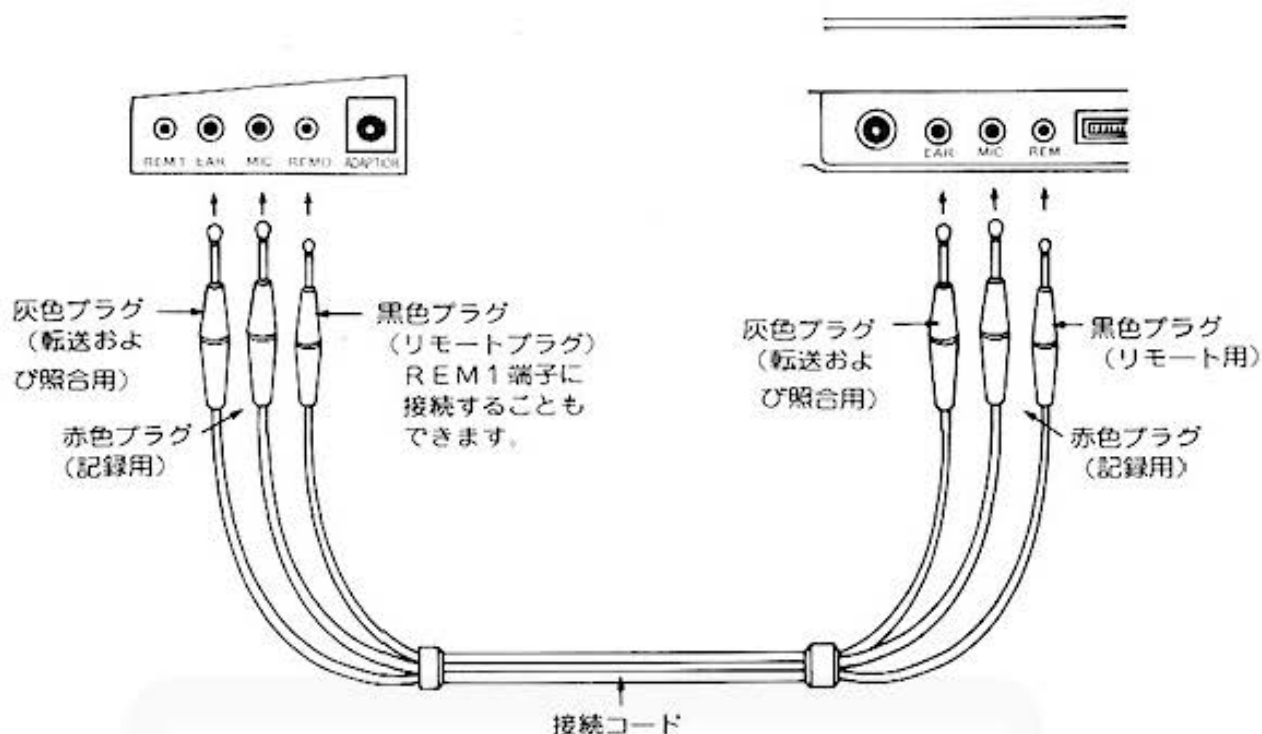
プログラムなどをテープに記録する、あるいはテープから転送する場合の計算機とテープレコーダーの操作方法について説明します。

注) テープレコーダーは別売のポケットコンピュータ専用カセットテープレコーダーCE-152を使用されているものとして説明します。

## (1) 準備

### ◎ 各機器の接続

- 計算機の **OFF** キーを押して電源をオフにし、計算機とCE-150を接続してください。(接続方法は82ページ参照)
- CE-150とCE-152を次のように接続してください。(テープレコーダーを1台使用する場合。)



- CE-152の取扱説明書も合わせてお読みください。

#### ◎ テープレコーダーに関する注意

- 転送あるいは照合をおこなうときは、記録をおこなったテープレコーダーを使用してください。記録したテープレコーダーと転送あるいは照合するテープレコーダーが異なる場合は転送あるいは照合できないことがあります。
- テープレコーダーのヘッドなどが汚れていますと、ひずみの増加、レベルの低下などがおこりますので、ヘッドの清掃をおこなってください。
- テープは一般に市販のものをご使用ください。ただし、極端に周波数特性の悪いもの、また、傷や折目があるものは使用しないでください。

## (2) テープへの記録方法

例としてプログラムを記録する方法について説明します。マニュアル操作の場合  
(プログラム)

```
10 INPUT "HANKET_R=":A
20 S=π*A*A
30 PRINT "MENSEKI_S=":S
40 END
```

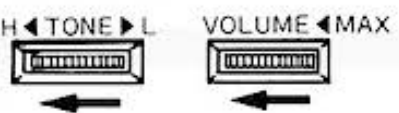
手 順	操 作		備 考
	RE M 0 使用の場合	RE M 1 使用の場合	
(1)	プリンタのリモートスイッチをOFF側にしてください。	RMT OFF <input type="button" value="ENTER"/> と押して、リモート機能を解除してください。	
(2)	テープをテープレコーダーにセットしてください。		テープの中の記録されていない部分をさがし、記録する位置をきめてください。 (テープの始めや終りのリーダーテープ部分には記録できませんので、さけてください。)
(3)	記録する内容を計算機に入れてください。 たとえば、前ページのプログラムを計算機に書き込んでください。		
(4)	プリンタのリモートスイッチを、ON側にしてください。	RMT ON <input type="button" value="ENTER"/> と押して、リモート機能をセットしてください。	
(5)	テープレコーダーの記録ボタン(●)を押し込んで、録音状態にしてください。		
(6)	記録命令を実行します。 プログラム： CSAVE "ファイル名" <input type="button" value="ENTER"/> データ： PRINT # "ファイル名" ; 変数, 変数, …… <input type="button" value="ENTER"/> (例)PROモードかRUN モードを指定 CSAVE "PR-1" <input type="button" value="ENTER"/>	記録命令を実行します。 プログラム： CSAVE -1 "ファイル名" <input type="button" value="ENTER"/> データ： PRINT #-1, "ファイル名" ; 変数, 変数, …… <input type="button" value="ENTER"/> (例)PROモードかRUN モードを指定 CSAVE -1 "PR-1" <input type="button" value="ENTER"/>	記録命令が実行されますと、まず最初に約 8 秒間無信号区間が記録(ピー音が記録)され、そのあとにファイル名とプログラムなどが記録されます。(プログラムなどの記録中はビビビビ……あるいは ビビビビ ……というような音が出ます。)
	記録が終了しますと、プロンプト表示になり、テープが止まります。 テープを巻きもどして、照合をおこなってください。 [ 巻きもどし、早送りなどをおこなうときは、 プリンタのリモートスイッチをOFF側にし て、おこなってください。 ]		照合については、101ページを参照ください。
	[ 巻きもどし、早送りなどをおこなうときは、 RMT OFF <input type="button" value="ENTER"/> と操作して、リモート機能を解除してからおこなってください。 ]		

- プログラムの実行において、データの記録をおこなう場合（PRINT # 命令）には、プログラム実行前にテープレコーダーの操作をしておいてください。

記録をおこなうとき、記録を始めたときのテープカウンタの数値を紙などに控えておけば、記録位置や、記録されていない位置を探し出すときに便利です。

### （３）計算機とテープの内容の照合

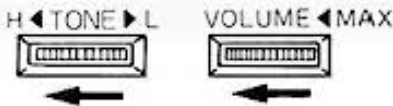
計算機に記憶されているプログラムあるいはリザーブ内容と、テープに記録されているプログラムあるいはリザーブ内容との照合を取る場合（マニュアル操作でのみ実行可能）には次のようにおこないます。

手 順	操 作		備 考
	REM 0 使用の場合	REM 1 使用の場合	
(1)	プリンタのリモートスイッチをOFF側にしてください。	RMT OFF <input type="button" value="ENTER"/> と押して、リモート機能を解除してください。	
(2)	照合する内容が記録されているテープをテープレコーダーにセットしてください。 そして、テープカウンターなどを見ながら、照合する内容の記録されている位置の近くまでテープを送ってください。		
(3)	プリンタのリモートスイッチをON側にしてください。	RMT ON <input type="button" value="ENTER"/> と押して、リモート機能をセットしてください。	
(4)	音量調整つまみ、音質調整つまみを図の矢印方向いっばいに回しておいてください。 <div style="text-align: center;">  </div>		
(5)	テープレコーダーの再生ボタン（▶）を押し込んで、再生状態にしてください。		
(6)	照合命令を実行します。 CLOAD? "ファイル名" <input type="button" value="ENTER"/> (例)PROモードかRUN モードを指定 CLOAD? "PR-1" <input type="button" value="ENTER"/> 両方の内容が一致している場合は実行を終了しプロンプト表示になります。	照合命令を実行します。 CLOAD?-1"ファイル名" <input type="button" value="ENTER"/> (例)PROモードかRUN モードを指定 CLOAD?-1"PR-1" <input type="button" value="ENTER"/>	ファイル名は記録をしたファイル名を入れてください。ファイル名を省略した場合はテープが回り出して最初の記録内容と、計算機内の内容との照合をおこないます。

もし、エラー表示 (ERROR 43) になった場合は、もう一度最初から照合をおこなってください。 それでもなお、エラーになる場合は、もう一度“記録”からおこなってください。	ファイル名やモード指定などにまちがいがいいか、よく確認しながら操作してください。
--	--

#### (4) テープからの転送

テープに記録されているプログラムあるいはリザーブ内容、データを計算機に転送する (テープより読み出す) 場合は次のようにおこなってください。(マニュアル操作の場合)

手 順	操 作		備 考
	REM 0 使用の場合	REM 1 使用の場合	
(1)	プリンタのリモートスイッチをOFF側にしてください。	RMT OFF <input type="button" value="ENTER"/> と押して、リモート機能を解除してください。	
(2)	転送する内容が記録されているテープをテープレコーダーにセットしてください。 そして、テープカウンターなどを見ながら、転送する内容の記録されている位置の近くまでテープを送ってください。		
(3)	プリンタのリモートスイッチをON側にしてください。	RMT ON <input type="button" value="ENTER"/> と押して、リモート機能をセットしてください。	
(4)	音量調整つまみ、音質調整つまみを図の矢印方向いっばいに回しておいてください。 <div style="text-align: center;">  </div>		
(5)	テープレコーダーの再生ボタン (▶) を押し込んで、再生状態にしてください。		
(6)	転送命令を実行します。 プログラム： CLOAD “ファイル名” <input type="button" value="ENTER"/> データ： INPUT # “ファイル名” ; 変数, 変数, ... <input type="button" value="ENTER"/>	転送命令を実行します。 プログラム： CLOAD-1 “ファイル名” <input type="button" value="ENTER"/> データ： INPUT #-1, “ファイル名” ; 変数, 変数, ... <input type="button" value="ENTER"/>	ファイル名は記録をしたファイル名を入れてください。  ファイル名を省略した場合はテープが回り出して最初に記録されている内容が転送されます。

<p>(例)PROモードかRUN モードを指定 CLOAD "PR-1" ENTER</p>	<p>(例)PROモードかRUN モードを指定 CLOAD-1 "PR-1" ENTER</p>	
<p>転送が終了すればプロンプト表示になります。</p>	<p>もし、途中でエラー (ERROR 43, ERROR 44) になった場合は、もう一度最初から転送をおこなってください。</p> <p>それでもなお、エラーになる場合は、もう一度"記録"からおこなってください。</p>	<p>ファイル名やモード指定などにまちがいがいいか、よく確認しながら操作してください。</p>

- プログラムの実行において、転送をおこなう場合 (CHAIN 命令、INPUT # 命令) には、プログラム実行前にテープレコーダーの操作をしておいてください。

#### ＜その他の注意＞

- ① 各機器の接続は確実にこない、連動中には抜かないでください。着脱は必ず計算機の電源をオフにしてからおこなってください。
  - ② AC電源や、ACアダプタでテープレコーダーを使用して、連動がうまくいかないとき、あるいは計算機が正常に動作しないときには、電池電源を使用してください。
- ACラインノイズのひどい環境で、AC電源でテープレコーダーを使用しますと、たとえば連動がうまくいかない、計算機が正常な動作をおこなわない (誤動作をする) などの現象が表われることがあります。(このようなとき、市販の雑音防止器を使用すると効果がある場合があります。)



## (5) CE-152以外のテープレコーダーを使用する場合について

















CE-152以外のテープレコーダーでも次のような条件を満たすれば使える場合があります。

























条 項	条 件
種 類	カセットテープ、マイクロカセットテープ、オープンリールテープを使用するテープレコーダーおよびテープデッキ
入力端子	MIC (マイク)端子あるいはそれに代る端子を有すること (ミニジャック) 注) AUX (補助入力)端子は使用不可
• 入力インピーダンス	上記端子が、インピーダンス $200\Omega \sim 1\text{K}\Omega$ 以下もしくはローインピーダンス用であること
• 最小入力レベル	3 mV 以下もしくは -50 dB 以下
出力端子	EXT. SPEAKER (外部スピーカー) 端子あるいはそれに代る MONITOR (モニター) 端子、EARPHONE (イアホン) 端子などを有すること (ミニジャック)
• 出力インピーダンス	上記端子のインピーダンスが $10\Omega$ 以下であること
• 出力レベル	1 V 以上 (実用最大出力 100 mW 以上であること)
歪率 (位相歪などを含む)	15% 以内 (同位相かつ 2 kHz ~ 4 kHz 内において)
ワウフラッター	0.3% 以内 (W. R. M. S.) 注) ワウフラッターはテープレコーダーに振動を与えたり、ショックを与えたりしますと、大きくなります。
そ の 他	極端な回転の変動がないこと


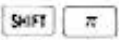






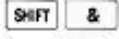



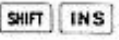

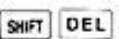
注) テープレコーダーの中には、接続条件が一部合わないもの、あるいは長期使用などにより、歪やノイズの増加、出力の低下など、電気的特性が変化して正常に動作しないことがあります。

## キーの機能

次に主なキーの機能を説明します。

キ ー	機 能
	<ul style="list-style-type: none"> <li>○電源を入れるときに押します。</li> <li>○プログラム実行中、プログラムを一時停止状態(BREAK状態……ブレイク)にします。</li> <li>○表示などをクリアします。(エラー状態を解除します。)</li> </ul>
	<ul style="list-style-type: none"> <li>○電源を切るときに押します。</li> </ul>
	<ul style="list-style-type: none"> <li>○π、ハ、?のように、キー上に直接記されずに、薄茶色でパネル上に記されている命令(機能)を指定します。</li> <li>(例)   → ? が入力されます。</li> <li>○アルファベットの小文字を入力するとき、アルファベットキーの前に押します。</li> </ul>
	<ul style="list-style-type: none"> <li>○入力中の内容や表示をクリアするときに押します。(エラー状態を解除します。)</li> </ul>
 	<ul style="list-style-type: none"> <li>○計算機の状態を解除します。(クリアオール) <ul style="list-style-type: none"> <li>●プログラムの実行が一時停止状態にあるとき、実行を中止させます。</li> <li>●WAIT 命令による時間指定を解除します。</li> <li>●表示フォーマット指定(USING 指定)を解除します。</li> <li>●トレースモードの解除</li> <li>●エラー状態の解除</li> <li>その他</li> </ul> </li> </ul>
	<ul style="list-style-type: none"> <li>○RUNモードとPROモードとの指定を切替えます。</li> </ul>
 	<ul style="list-style-type: none"> <li>○RUNモードとRESERVEモードとの指定を切替えます。</li> </ul>
 ~ 	<ul style="list-style-type: none"> <li>○数値を指定します。</li> </ul>
	<ul style="list-style-type: none"> <li>○小数点を指定します。</li> <li>○命令入力の際の省略形の指定をおこないます。</li> <li>○USING 命令内において、数値データの表示に小数点以下の表示をおこなう指定をします。</li> </ul>
	<ul style="list-style-type: none"> <li>○指数部入力を指定します。(このキーはアルファベットキーと同じです。)</li> </ul>
	<ul style="list-style-type: none"> <li>○除算命令を指定します。</li> </ul>

キ ー	機 能
	<ul style="list-style-type: none"> <li>○乗算命令を指定します。</li> <li>○USING命令内でも使用します。</li> </ul>
	<ul style="list-style-type: none"> <li>○数値の符号（正符号）を指定します。（一般に省略可）</li> <li>○加算命令を指定します。</li> </ul>
	<ul style="list-style-type: none"> <li>○数値の符号（負符号）を指定します。</li> <li>○減算命令を指定します。</li> </ul>
 	○CLOAD?命令を指定するときに使用します。
 	○1ラインに複数のステートメントを定義するときの区切りを指定します。
 	<ul style="list-style-type: none"> <li>○式と式の区切りや変数の区切り、コメントの区切りなどを指定します。</li> <li>○USING命令内で、数値の3桁区切りマークをつけて表示をおこなう指定をします。</li> </ul>
 	○式や変数の区切り、コメントの区切り、命令文と変数の区切りなどを指定します。
	<ul style="list-style-type: none"> <li>○代入文において、右辺の内容を左辺で指定される変数に代入することを指定します。</li> <li>○IF文の条件式などに使用します。</li> </ul>
 	<ul style="list-style-type: none"> <li>○固定変数を添字付変数として指定するときに使用します。 (例) @ (1)、@ \$ (2) など</li> <li>○ソフトキーを定義付けキーとして使用する場合などに、リザーブ内容として使用します。 (例) GOTO 100@ など</li> </ul>
	<ul style="list-style-type: none"> <li>○A、S、D、…などのラベルで定義されたプログラムの実行を開始するとき、そのラベルに対するキーを押す前に押します。(Definable キー指定)</li> <li>○Q、W、E、…などのキーに定義されている命令を入力するとき、これらのキーの前に押します。</li> </ul>
 ~ 	<ul style="list-style-type: none"> <li>○アルファベットキーで、各命令や、変数名などを入力します。</li> <li>○DEF キーに続いて押せば定義付けキーとして働きます。</li> <li>○SHIFT キーに続いて押せば小文字が入力されます。</li> </ul>
 , 	○カッコの指定をおこないます。
   	○IF文の条件式などに使用します。
	○プログラム入力あるいは文字列入力時にスペースを設けるための命令です。 スペースはプログラム、演算などの実行時には無視されます。

キ ー	機 能
	<ul style="list-style-type: none"> <li>○べき乗命令を指定します。</li> <li>○USING 命令内において、数値データの表示形式を浮動小数点表示（指数表示）形式に指定します。</li> </ul>
	<ul style="list-style-type: none"> <li>○<math>\pi</math>（パイ）を指定します。</li> </ul>
	<ul style="list-style-type: none"> <li>○開平計算の命令を指定します。</li> </ul>
	<ul style="list-style-type: none"> <li>○表示を、現在の内容と書き込まれているキーシンボル（ソフトキーのキーシンボル）と入れ替えます。</li> </ul>
	<ul style="list-style-type: none"> <li>○ソフトキーにリザーブされている内容のリザーブナンバー（Ⅰ、Ⅱ、Ⅲ）を切替えます。</li> </ul>
	<ul style="list-style-type: none"> <li>○オプション（別売）のカタカナがあるときに、カナ入力モードの指定と解除をおこないます。</li> </ul>
	<ul style="list-style-type: none"> <li>○リザーブされている内容呼び出すとき、あるいはリザーブをおこなうときに押します。（ソフトキー）</li> <li>○リザーブされていないキーを押したときは、それぞれのキーが持つキャラクターシンボル（! " # \$ % &amp;）が入力されます。</li> </ul>
 	<ul style="list-style-type: none"> <li>○それぞれのキャラクターシンボルを入力します。</li> <li>”：文字列の指定をおこないます。</li> <li>ラベルの指定をおこないます。</li> <li>#：USING 命令内において、数値データの表示フォーマットを定義します。PRINT # 命令および INPUT # 命令を指定するときに使用します。</li> <li>\$：文字変数を指定します。</li> <li>&amp;：USING 命令内において、文字列の表示フォーマットを定義します。</li> <li>16進数値を示します。</li> <li>! % : * ^ の中で文字列として使用します。</li> </ul>
	<ul style="list-style-type: none"> <li>○カーソルの右シフトをおこないます。</li> <li>○プレイバック命令の実行をおこないます。</li> <li>○プログラムおよびリザーブ内容が表示されているとき、カーソルが表示されていない場合にカーソルを呼び出します。（コロンの右隣）</li> <li>○マニュアル操作時のエラーを解除します。</li> </ul>
	<ul style="list-style-type: none"> <li>○カーソルの左シフトをおこないます。</li> <li>○そのほか  キーと同じ。</li> </ul>
	<ul style="list-style-type: none"> <li>○カーソルが示す位置に、1ステップ容量の挿入スペース（ただし  表示）を設けます。</li> </ul>
	<ul style="list-style-type: none"> <li>○カーソルが示す位置の内容を削除します。</li> </ul>

キ ー	機 能
<b>ENTER</b>	<ul style="list-style-type: none"> <li>○プログラムのラインの終了命令を指定します。</li> <li>○プログラム、リザーブ内容、キーシンボルなどの書き込みをおこないます。</li> <li>○マニュアル計算の実行あるいはBASIC命令のマニュアル操作による実行を命令します。</li> <li>○INPUT 命令や PRINT 命令で、一時停止しているプログラムの再スタートを命令します。</li> </ul>

**↓**、**↑** キーについては、モードの指定および計算機の状態により、次のように働きます。

モ ー ド	状 態	<b>↓</b>	<b>↑</b>
RUN	プログラム実行中		
	プログラムの一時停止中 INPUT 命令実行時 PRINT 命令実行時 ブレーク状態	次のラインを実行（1ラインずつ実行して停止します。）	押している間、実行しているラインあるいは実行したラインを表示
	プログラム実行時のエラー状態		押している間、エラー発生 of ラインを表示
	トレースモードオン状態	デバッグ動作を実行	押している間、実行しているラインあるいは実行したラインを表示
PRO（他のモードから PRO モードに切換え、プログラムラインが表示されていないとき）			
	プログラムの一時停止中	実行を停止しているラインを表示	同 左
	エラー発生後 <b>ON</b> あるいは <b>CL</b> でエラーを解除したとき。	エラー発生 of プログラムラインを表示	同 左
	そ の 他	先頭ラインを表示	末尾ラインを表示
PRO（プログラムラインが表示されているとき）			
		次のプログラムラインを表示	1ライン前のラインを表示
RESERVE			

- **ENTER** キーは表示上ではスペースと同じです。
- **OFF** キーはプログラムの実行中など、内部で演算処理をおこなっているときは働きません。
- キー入力待ちの状態、約7分間キー入力がないと自動的に電源が切れます。  
(Auto power off: オートパワーオフ機能)

## ◎ 表示シンボル

表示部の上側に表示されるシンボルの意味を説明します。

**BUSY** : プログラムを実行していることを示します。

**SHIFT** : SHIFT キーが押されたことを示します。このシンボルが表示されているときは、キーの上側のパネルに薄茶色で記されている機能が指定されています。

**カナ** : オプションの「カタカナ」を持っている場合に、カナが入力できる状態になっていることを示します。

**DEG** : ディグリー [°]

**RAD** : ラディアン [rad]

**GRAD** : グラード [g]

三角関数や逆三角関数の計算において、  
取り扱う角度の単位を示すシンボルです。

$$(90^\circ = \frac{\pi}{2} [\text{rad}] = 100^g)$$

**RUN** : ランモード

**PRO** : プログラムモード

**RESERVE** : リザーブモード

計算機のモード(状態)を示すシンボルです。

**DEF** : DEF キーが押されたことを示します。このシンボルが表示されているとき、アルファベットキーなどが定義付けキーや割り当てられている命令の入力キーとして働きます。

**I**

**II**

**III**

リザーブナンバー

ソフトウェアキーの何番目の内容かを示します。

● : バッテリーインジケーター

電源を入れたとき、電池の電圧が十分にある場合は表示部の右はしにグレー(灰色)のバッテリーインジケーター「●」が表示されます。(他のシンボルなどに比べ、わずかにうすく表示されます。)

電池の電圧が規定より低くなりますと「●」が正面から見えなくなります。



## オプション機器のご案内

PC-1500の別売周辺機器として次のものが用意されています。

### ●CE-150

カラーグラフィックプリンタ(カセットインターフェイス内蔵)

CE-150はカセットインターフェイスを内蔵した4色カラーグラフィックプリンタです。プリンタはボールペン方式で、黒、青、緑、赤の4色のペンを取りつけることができ、プログラムにより任意の場所から直線、点線などをそれぞれの色で書くことができます。これらの色や線を組み合わせれば、カラフルなグラフや任意の図形を書くことができます。

もちろんプログラムのリスト、データの自動印字などもできます。

CE-150にはテープレコーダーを2台接続することができ、一方を記録用、もう一方を転送用として同時に使用することができます。たとえば、前日のデータファイルを転送しながら今日のデータを集計し、記録していくことができます。

### ●CE-151

メモリーモジュール

CE-151は4KバイトのRAMモジュールです。PC-1500本体に取りつけることによりプログラム・データエリアを拡張することができます。

○CE-151をPC-1500に取りつけることにより、プログラム・データエリアの容量は5946バイトになります。

### ●CE-152

ポケットコンピュータ専用カセットテープレコーダー

CE-152はポケットコンピュータ専用のカセットテープレコーダーです。CE-152をPC-1500の外部記憶装置として、テープに大切なプログラムやデータを記録しておけば、必要なときに計算機に読み込んで利用することができます。

なお、CE-152を使用する場合は、カラーグラフィックプリンタCE-150が必要です。

# 仕 様

形 名……P C -1500

計 算 桁 数……10桁（仮数部）+ 2桁（指数部）

計 算 方 式……数式通り（優先順位判別機能つき）

プログラム言語……B A S I C（ベーシック）

内部システム……CPU：CMOS 8ビットCPU

構 成 ROM：16K バイト

RAM：3.5K バイト

システムエリア 0.9K バイト

入力バッファ 80バイト

スタック 196バイト

その他

ユーザーエリア 2.6K バイト

データ専用エリア(A～Z, A\$～Z\$) 624バイト

プログラム・データエリア 1850バイト

リザーブエリア 188バイト

基本計算機能……基本計算：加減乗除算

関数計算：三角関数、逆三角関数、対数、指数、角度変換、べき乗、開平方、計算、整数化、絶対値、符号関数、円周率

編集機能……カーソル左右シフト（▶, ◀）

挿入（INS）

削除（DEL）

ラインアップ・ダウン（↓, ↑）

メモリー保護……バッテリーバックアップ

（プログラム、データ、リザーブ内容など電源オフ時内容保護）

表 示……ミニグラフィック表示：7×156ドット

文字表示：26桁

使用温度……0℃～40℃

電 源……6V ー（DC）：単3タイプ乾電池4本

電池使用時間……単3タイプ乾電池（SUM-3） 約50時間

消費電力……0.13W

外形寸法……幅195mm×奥行86mm×高さ25.5mm

重 量……375g（電池含む）

付 属 品……ソフトケース、テンプレート2枚、単3乾電池4本、取扱説明書、  
ネームラベル、シャープサービス・お客様ご相談窓口所在地一覧表

## ▶ ポケットコンピュータ友の会御案内 ◀

### 入会手続

1. 皆様の作成されました興味あるオリジナルプログラム一編以上を巻末の所定フォーム（のコピー）を用いて、本取扱説明書中のプログラム事例集に従がい記載いただき、下記送付先へお送り下さい。

（注）御送付頂いたプログラムは返却致しませんので御了承ください。また御送付頂いたプログラムの製本化やカセットテープ化等に伴う編集及び出版販売等に関する一切の権利は弊社（シャープ株）に無償で譲度頂きます。プログラムは、オリジナルなもので、弊社（シャープ株）以外への2重投稿はお断わり致します。

### 会員特典

1. 会員証を発行致します。（有効期間1年）
1. 御送付頂いたプログラムは、弊社（シャープ株）で選考の上、製本化し、出版会社を通じて出版・販売致します。尚、採用プログラム分につきましては記念品を贈呈させて頂きます。
1. 上記の本を作る際の該当期間中に、オリジナルプログラムをお送り頂いた会員の皆様には、この本を無償で贈呈致します。

送付先 〒639-11 奈良県大和郡山市美濃庄町492  
シャープ株式会社 産業機器事業本部  
電卓事業部 ポケットコンピュータグループ

## プログラム事例集

このページまで各種機能の説明を読み進まれ、多くのプログラム命令の知識を習得されたと思います。しかしながら、BASIC言語で自由自在にプログラム作成ができるようになるには、この取扱説明書に加え、実際のプログラムを自分で作成して試みる必要があります。車の運転やテニスなどと同じで、自分自身で実際にハンドルをにぎり、ラケットを振って練習することにより、始めて上達するように、プログラムも上手、下手を気にせずに、数多く作成することにより始めて上達します。他人の作成したプログラムを参考にすることも大切です。

以降のページにBASICの各命令をいろいろ使用したプログラムをいくつか載せていますので参考にしてください。

(なお、当社はこのプログラムの使用に際して生ずる金銭上の損害および逸失利益については、一切の責任を負いませんのであらかじめご了承ください。)

### プログラムライブラリーのご案内

#### 【ポケットコンピュータプログラムライブラリー Series-2 (PC-1500用)】

事務分野や数学統計・技術さらにはゲーム分野などで実用性と楽しさ、また、はん用性に優れたプログラム69本を編集し、昭和57年初めより出版社を通じて出版・販売致しております。

くわしくはお近くの販売店または有名書店にお問い合わせください。

---

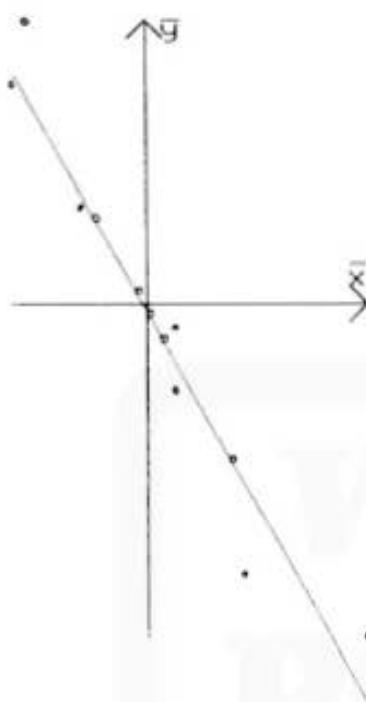
# 目 次

---

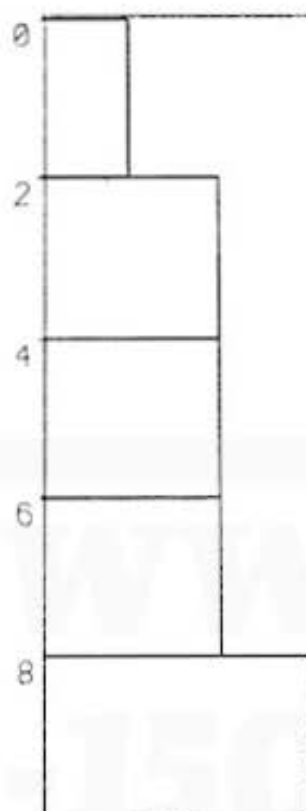
(プログラム名)	(ページ)
●方程式の根.....	122
●相関係数, 1 次回帰, プロット.....	126
●借入限度額、ローン返済回数の計算.....	133
●並びかえ.....	136
●度数分布図.....	141
●バイオリズム.....	146
●タイプ練習.....	151
●ストップウォッチ, タイマー, 目覚し時計.....	154
●コンピュータ・フラワー.....	160
●コンピュータ・グラフィック.....	163

## プリント出力例

### ★相関係数、1次回帰、 プロット (P.127 参照)



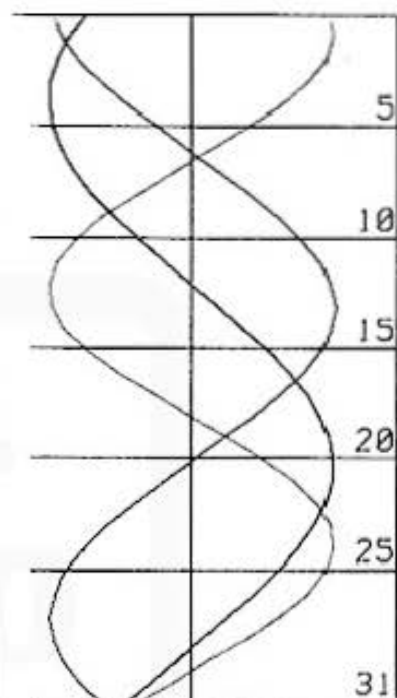
### ★度数分布図 (P.142 参照)



### ★バイオリズム (P.147 参照) (身体、感情、知性)

-- SHINTAI  
-- KANJYO  
-- CHISEI

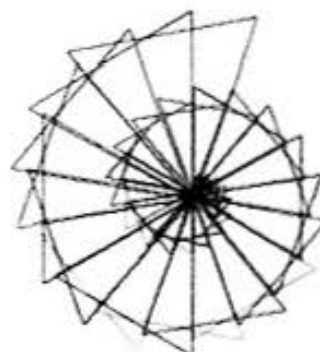
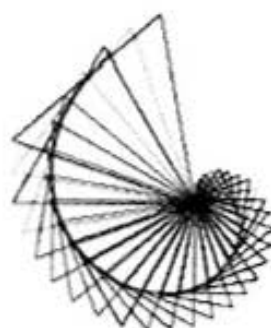
(-) (+)



### ★コンピュータ・フラワー (P.160 参照)



### ★コンピュータ・グラフィック (P.164 参照)



PROGRAM  
TITLE

## 方程式の根

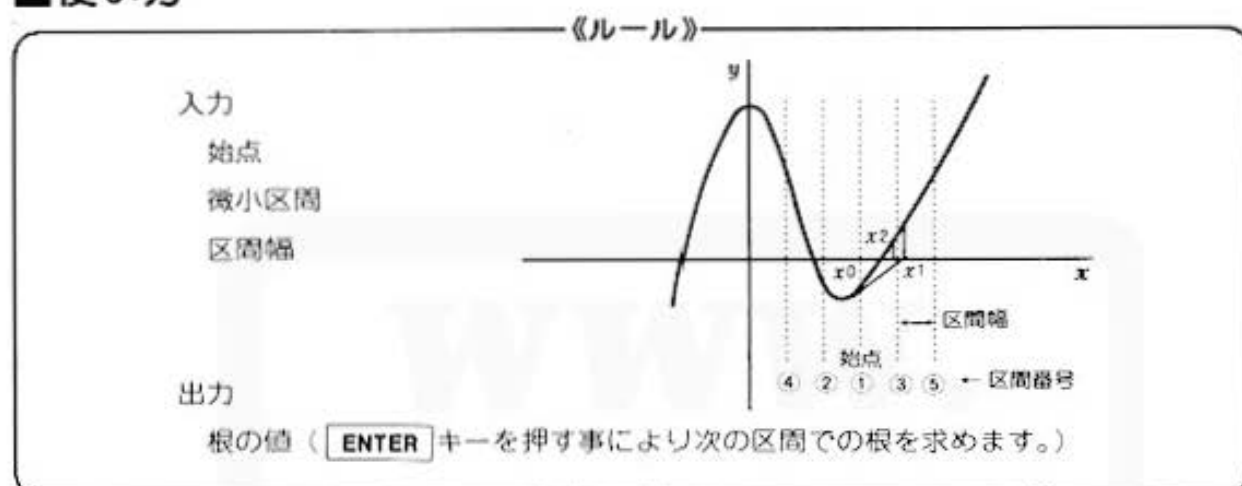
1

## 概要(数学分野)

一般の方程式の根を求めるのはなかなか面倒なものですが、ここではニュートン法を用いて根を近似して行く方法を紹介しています。

一つの根が求まると指定した区間幅に応じてニュートン法を用いる始点が自動的に変わっていきます。例題として3次方程式を探り上げてみました。

## ■使い方



## ■例題

$$x^3 - 2x^2 - x + 2 = 0 \quad (\text{根は } -1, 1, 2)$$

始点 = 0

微小区間 =  $10^{-4}$

区間幅 = 0.5

として計算します。

関数は、サブルーチンとして、500ライン以後に書き込んで下さい。

書き込み方(例題の場合) 1. PROモードにする。( **MODE** キーを押して )

2.  $500B=((X-2)*X-1)*X+2$  **ENTER**

510RETURN **ENTER** これで書き込み終了。

## ■内容 (計算内容など)

$$X_{n+1} = X_n - \frac{f(X_n)}{f'(X_n)}$$

$X_n$  と  $X_{n+1}$  の差の絶対値が  $10^{-8}$  以下になれば  $X_n$  を根とみなし表示します。

ここでは1次微分の  $f'(X)$  を次の様に定義します。

$$f'(X) = \frac{f(X+h) - f(X)}{h} \quad (h \text{ は微小区間})$$

$10^{-8}$  を変更する時は、340ラインの  $1E-8$  を変更して下さい。

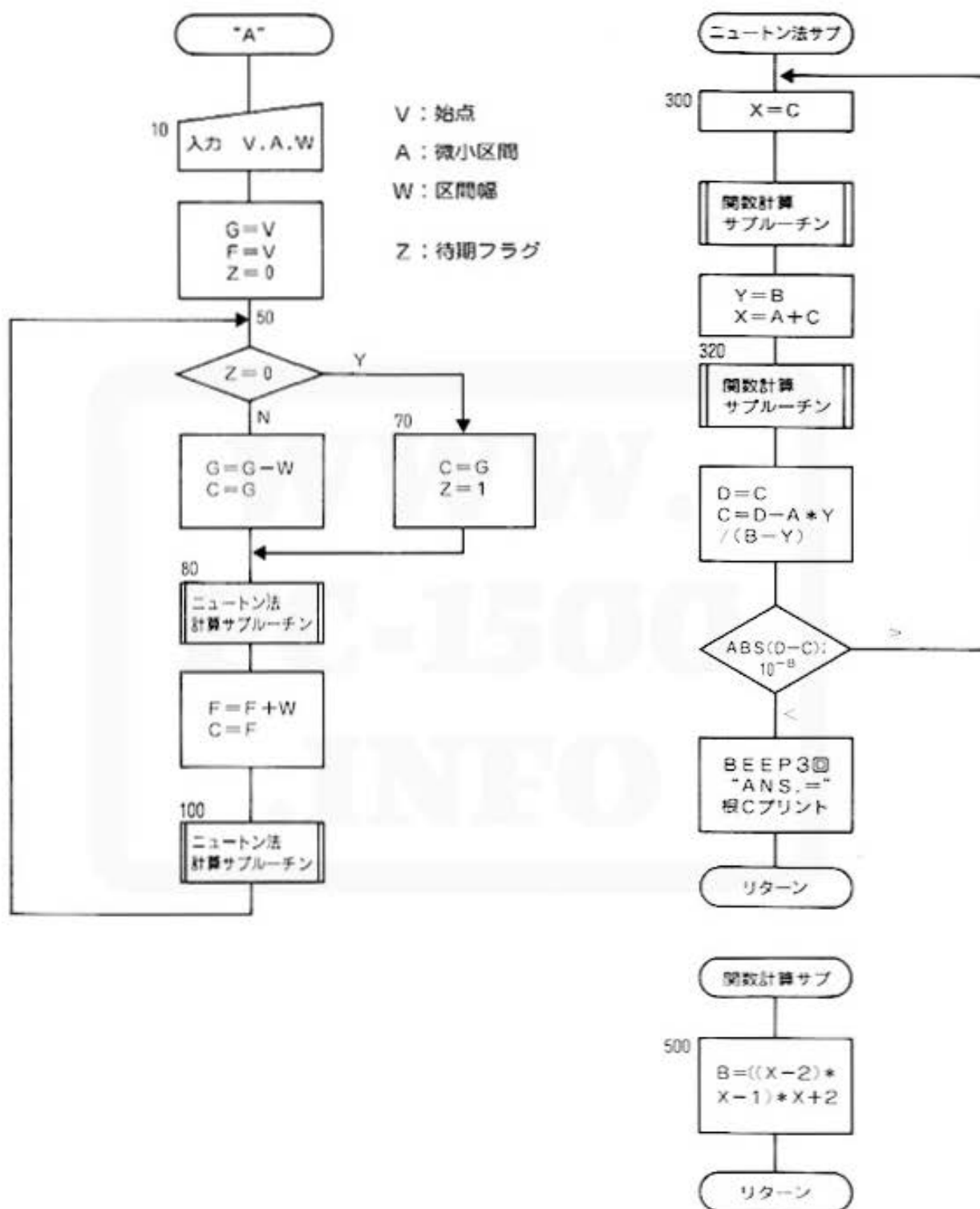


PROGRAM  
TITLE

## 方程式の根

2

## ■フローチャート



PROGRAM  
TITLE

## 方程式の根

3

## ■キー操作手順

ステップ	入 力	表 示	備 考
1	DEF A	SHITEN=	
2	0 ENTER	BISHO=	
3	0.0001 ENTER	KUKAN=	
4	0.5 ENTER	ANS.=2	
5	ENTER	ANS.=1	ENTER を繰り返し複数の根を求めます。
6	ENTER	ANS.=-1	}
	ENTER	ANS.=1	
	ENTER	ANS.=-1	
	ENTER	ANS.=-1	
	ENTER	ANS.=-1	
	ENTER	ANS.=2	
		ANS.=-1	

PROGRAM  
TITLE

## 方程式の根

4

## ■プログラムリスト ※文字の大きさは18字/行

```

10: "A": INPUT "SHI
    TEN="; U
20: INPUT "BISHO="
    ; A
30: INPUT "KUKAN="
    ; W
40: G=U: F=U: Z=0
50: IF Z=0 GOTO 70
60: G=G-W: C=G: GOTO
    80
70: C=G: Z=1
80: GOSUB 300
90: F=F+W: C=F
100: GOSUB 300
110: GOTO 50
120: END
300: X=C: GOSUB 500
310: Y=B: X=A+C
320: GOSUB 500
330: D=C: C=D-A*Y/(B
    -Y)
340: IF ABS (D-C)>=
    1E-8 GOTO 300
350: BEEP 3: PRINT "
    ANS. ="; C
360: RETURN
500: B=((X-2)*X-1)*
    X+2
510: RETURN

```

メモリー内容	
A	微小区間(入力値)=h
B	f(X)
C	X <sub>0</sub>
D	f(X+h)
E	
F	✓
G	✓
H	
I	
J	
K	
L	
M	
N	
O	
P	
Q	
R	
S	
T	
U	
V	始点(入力値)
W	区間幅(入力値)
X	X
Y	f(X)
Z	初期フラグ

PROGRAM  
TITLE

## 相関係数, 1次回帰, プロット

1

カラーグラフィック  
プリンター(CE-150)必要

## 概要(統計分野)

データは分析され、推定されるためにある。

このプログラムは2つの関連したデータ( $X_1, Y_1$ )...( $X_n, Y_n$ )の間の共分散と相関係数を求め、更に1次回帰式も求めます。その後与えられたデータに対して $Y = AX + B$ へのあてはめ(推定)を行ない、その結果もプリンターにグラフ化します。

## ■使い方

1. データの入力( $X_i, Y_i$ )(但し本体標準容量で $i \leq 10$ )
2. 共分散、相関係数、回帰係数、平均値を求めプリンターに出力。
3.  $\bar{X}$ ,  $\bar{Y}$ をX軸、Y軸とするグラフに表現し、入力データ及び推定値は色を変えてプロットします。
4. 推定値XからYを求め、X, Yの値をプリンターに出力します。

## ■例題

X	6.9	7.6	7.6	9.0	8.1	6.5	6.4	6.9
Y	12	10	9	5	6	15	14	12

分散 = -3.060714286

推定値 X = 7, Y = 11.85326587

相関係数 = -9.693968513E-01

X = 8, Y = 7.911223556

回帰係数 a = -3.942042318

X = 7.5, Y = 9.882244715

b = 39.4475621

X = 7.3, Y = 10.67065318

平均値  $\bar{X}$  = 7.375

X = 7.4, Y = 10.27644895

 $\bar{Y}$  = 10.375

.....

PROGRAM  
TITLE

## 相関係数, 1次回帰, プロット

2

## ■内容 (計算内容など)

$$Sxx = \sum x_i^2 - n\bar{x}^2$$

$$Sxy = \sum x_i y_i - n\bar{x}\bar{y}$$

$$Syy = \sum y_i^2 - n\bar{y}^2$$

$$C = Sxy / (n-1) \dots\dots\dots \text{共分散}$$

$$r = Sxy / \sqrt{Sxx Syy} \dots\dots \text{相関係数}$$

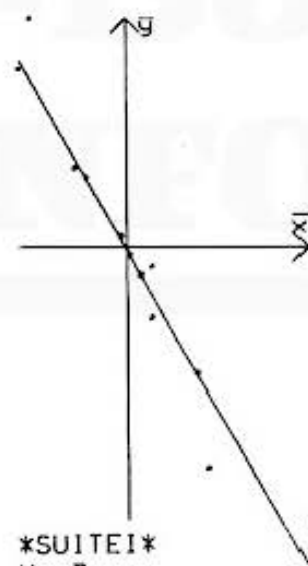
$$a = Sxy / Sxx \quad \left. \begin{array}{l} \\ \\ \end{array} \right\} \text{回帰係数 (} y = ax + b \text{)}$$

$$b = \bar{y} - a\bar{x}$$

## ■プリント出力 ※文字の大きさは18字/行

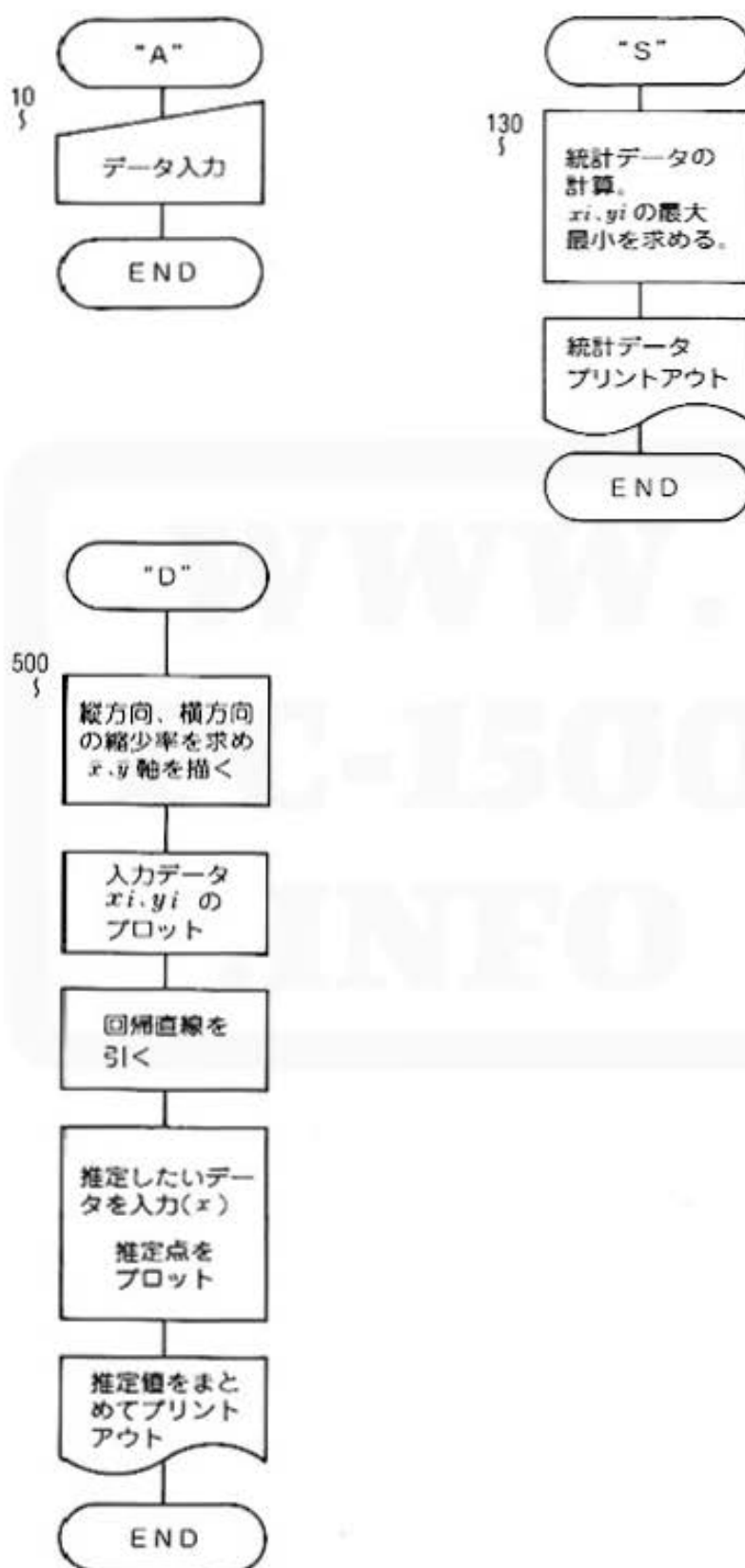
(実際はカラープリント出力です。)  
(P.121参照)

BUNSAN=  
-3.060714286  
SOUKAN=  
-9.693968513E-01  
KAIKI-KEISUU  
A=-3.942042318  
B= 39.4475621  
\*HEIKIN\*  
X= 7.375  
Y= 10.375



\*SUITEI\*  
X= 7  
Y= 11.85326587  
X= 8  
Y= 7.911223556  
X= 7.5  
Y= 9.882244715  
X= 7.3  
Y= 10.67065318  
X= 7.4  
Y= 10.27644895

## ■概略フローチャート

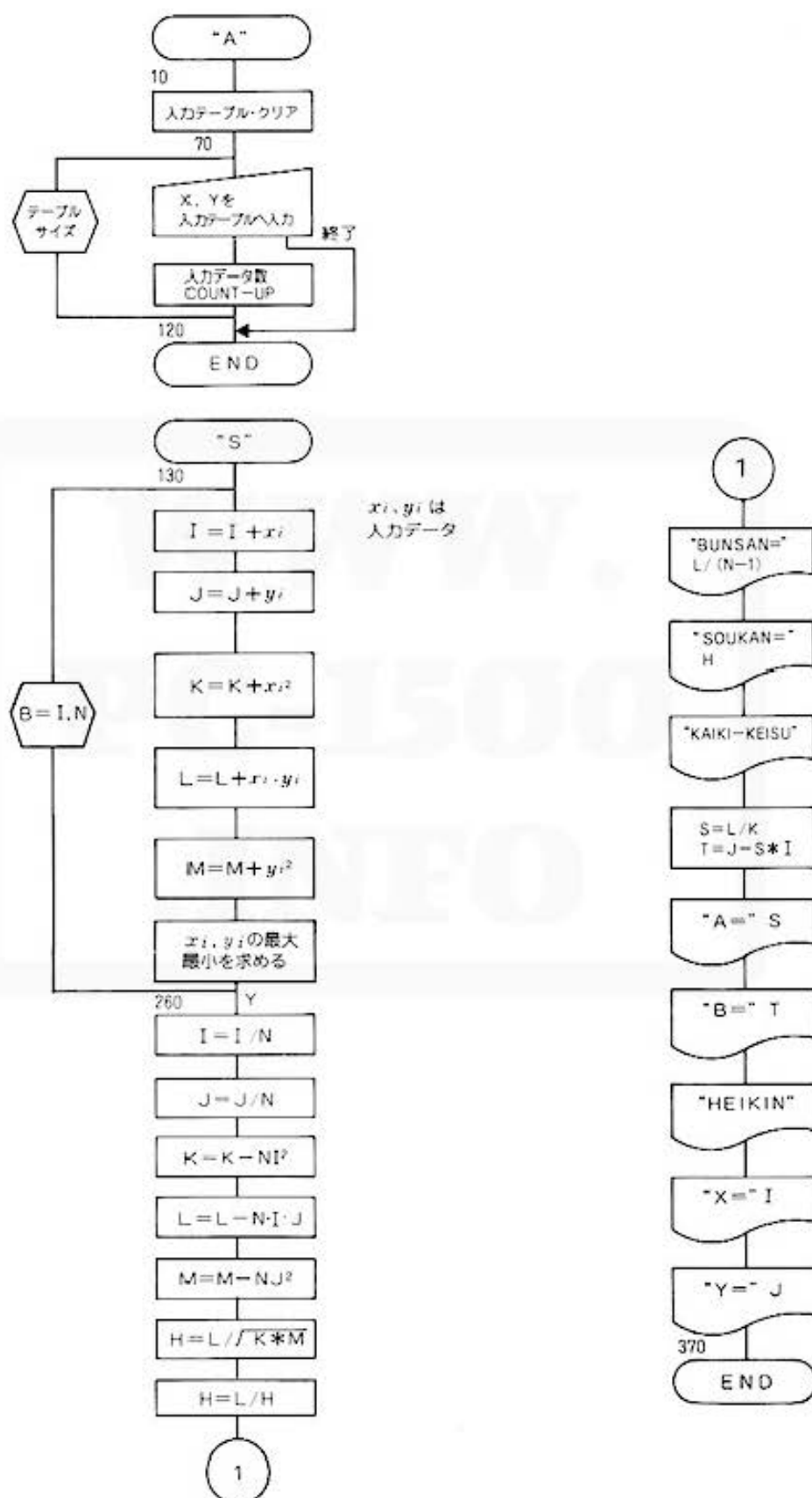


PROGRAM  
TITLE

## 相関係数, 1次回帰, プロット

3

## ■フローチャート



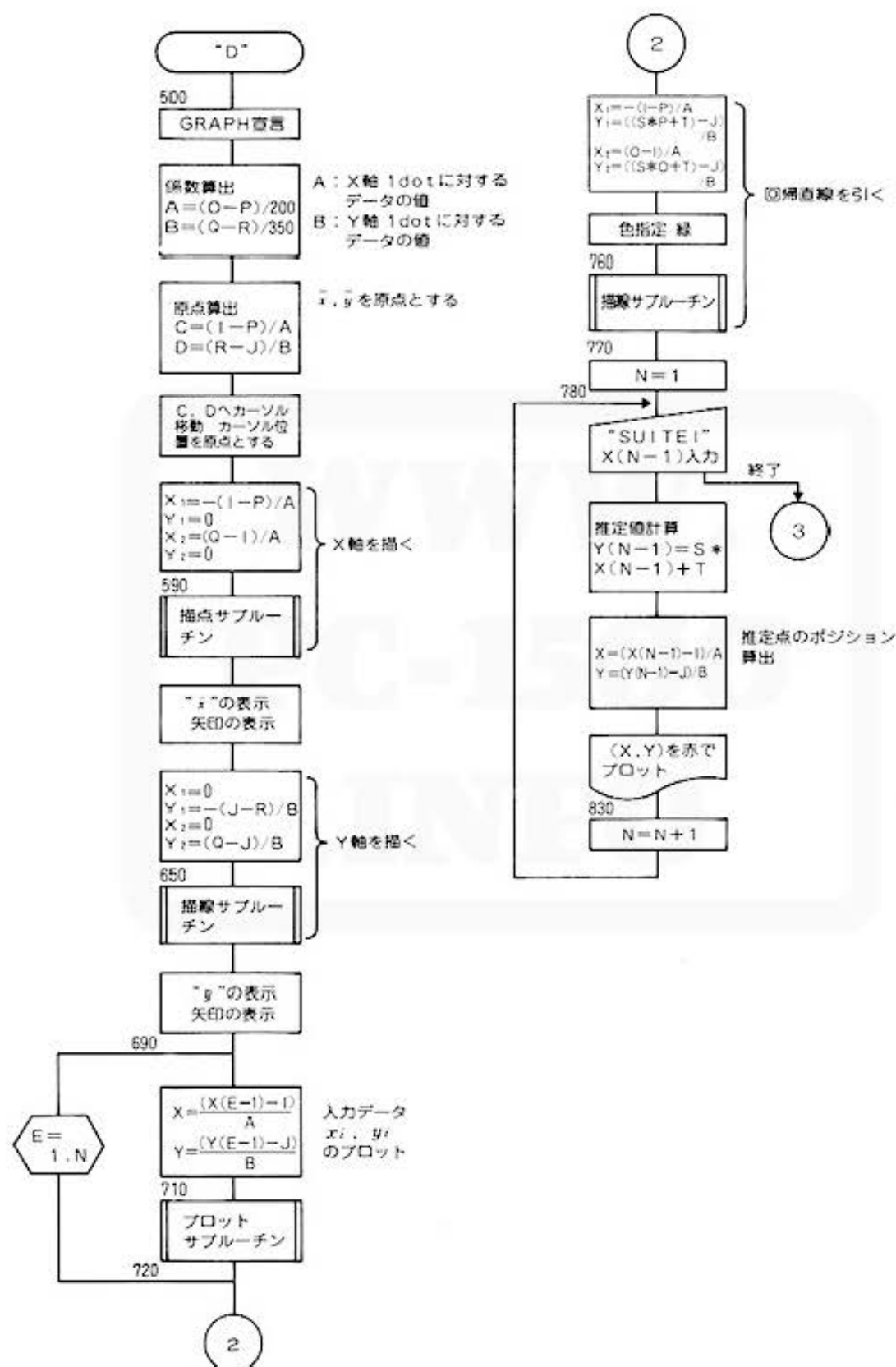


PROGRAM  
TITLE

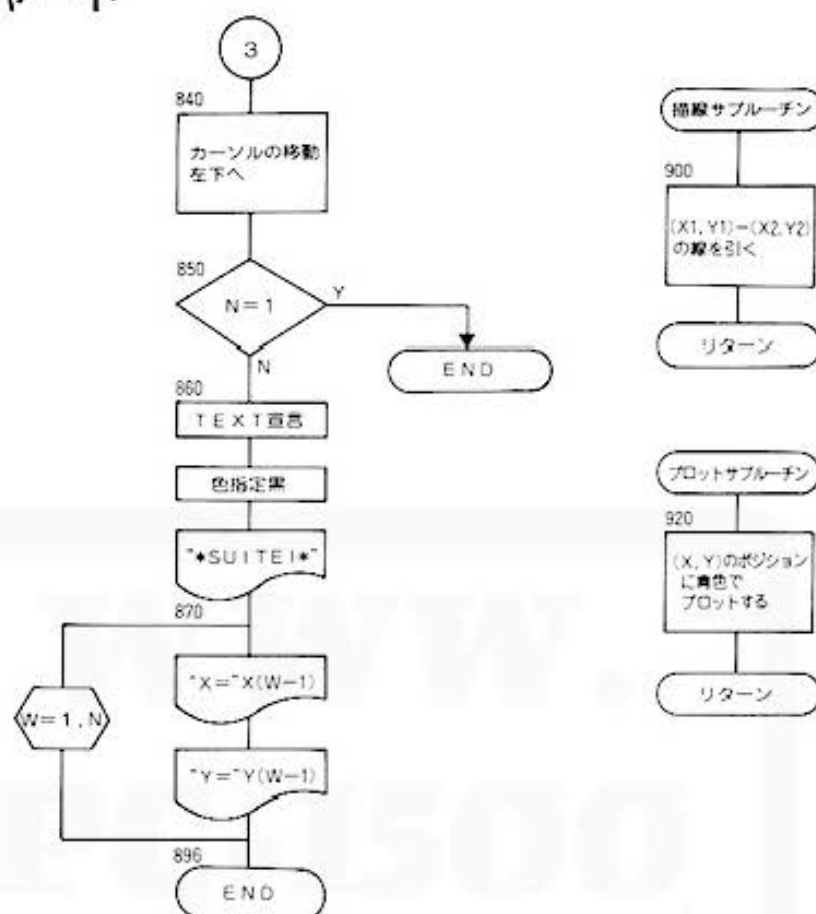
## 相関係数, 1次回帰, プロット

3

## ■フローチャート



## ■フローチャート



## ■キー操作手順

ステップ	キー入力	表示	備考
1	DEF A	X =	
2	6.9 ENTER	Y =	(1)の表示に戻り(2)で ENTER キーを押すか、又は10組 分のデータを入力する迄繰返しま す。
3	12 ENTER	X =	
⋮	⋮	⋮	
18	ENTER	>	
19	DEF S	>	>が表示されれば終了。この間にプ リンターに分散等を出力します。
20	DEF D	SUITEI =	表示が出る迄にプリンターにグラフ を出力します。
21	? ENTER	SUITEI =	(20)の表示に戻り10データ分入力す るか、又は ENTER のみ入力する 迄繰返します。
⋮	⋮	⋮	

PROGRAM  
TITLE

## 相関係数, 1次回帰, プロット

4

## ■プログラムリスト ※文字の大きさは18字/行

```

10:"A":CLEAR
20:DIM X(9),Y(9)
30:FOR B=1TO 10
40:X(B-1)=0:Y(B-1)=0
50:NEXT B
60:N=0
70:FOR B=1TO 10
80:INPUT "X=":X(B-1):GOTO 95
90:GOTO 120
95:INPUT "Y=":Y(B-1)
100:N=N+1
110:NEXT B
120:END
130:"S":I=0:J=0:K=0:L=0:M=0
140:P=10^(98):Q=-P:R=P:Q=0
150:FOR B=1TO N
155:Z=B-1
160:I=1+X(Z)
170:J=J+Y(Z)
180:K=K+X(Z)*X(Z)
190:L=L+X(Z)*Y(Z)
200:M=M+Y(Z)*Y(Z)
210:IF P>X(Z)LET P=X(Z)
220:IF Q<X(Z)LET Q=X(Z)
230:IF R>Y(Z)LET R=Y(Z)
240:IF Q<Y(Z)LET Q=Y(Z)
250:NEXT B
260:I=I/N:J=J/N
270:K=K-N*I*I
280:L=L-N*I*J
290:M=M-N*J*J
300:H=J/(K*M)
307:H=L/H
310:COLOR 0:LPRINT "BUNSAN=",L/(N-1)
320:LPRINT "SOUKAN=",H
330:LPRINT "KAIKI-KEISUU"
340:S=L/K:T=J-S*I
350:LPRINT "A=":S
360:LPRINT "B=":T
362:LPRINT "*HEIKI N*"
364:LPRINT "X=":I
366:LPRINT "Y=":J
370:END
500:"D":GRAPH
510:A=(Q-P)/200
520:B=(Q-R)/350
530:C=(I-P)/A
540:D=(R-J)/B
550:GLCURSOR (C,D)
560:SORGN
570:X1=-(I-P)/A:Y1=0
580:X2=(Q-I)/A:Y2=0
590:GOSUB 900
600:LINE (X2-10,Y2-10)-(X2,Y2)
605:LINE (X2,Y2)-(X2-10,Y2+10)
610:LPRINT "x"
620:LINE (X2-10,Y2+23)-(X2,Y2+23)
630:X1=0:Y1=-(J-R)/B
640:X2=0:Y2=(Q-I)/B
650:GOSUB 900
660:LINE (X2-10,Y2-10)-(X2,Y2)
665:LINE (X2,Y2)-(X2+10,Y2-10)
670:LPRINT "y"
680:LINE (X2+10,Y2)-(X2+20,Y2)
690:FOR E=1TO N
700:X=(X(E-1)-I)/A:Y=(Y(E-1)-J)/B
710:GOSUB 920
720:NEXT E
730:X1=-(I-P)/A:Y1=-(S*P+T)-J)/B
740:X2=(Q-I)/A:Y2=-(S*Q+T)-J)/B
750:COLOR 2
760:GOSUB 900
770:N=1
780:INPUT "SUITE1=":X(N-1):GOTO 800
790:GOTO 840
800:Y(N-1)=S*X(N-1)+T
810:X=(X(N-1)-I)/A:Y=(Y(N-1)-J)/B
820:LINE (X-1,Y-1)-(X+2,Y+2),0,3,B
830:N=N+1:GOTO 780
840:GLCURSOR (-(I-P)/A,-(J-R)/B-20)
850:IF N=1END
852:TEXT
860:COLOR 0:LPRINT "*SUITE1*"
870:FOR W=1TO N-1
880:LPRINT "X=":X(W-1)
890:LPRINT "Y=":Y(W-1)
895:NEXT W
896:END
900:LINE (X1,Y1)-(X2,Y2)
910:RETURN
920:LINE (X,Y)-(X+2,Y+2),0,1,B
930:RETURN

```

## メモリー内容

A	グラフ係数(1dot移動)X
B	〃 Y
C	✓
D	✓
E	✓
F	
G	
H	相関係数
I	$\bar{X}$
J	$\bar{Y}$
K	$S_{xx} = \sum X_i^2 - n\bar{X}^2$
L	$S_{xy} = \sum X_i \cdot Y_i - n\bar{X}\bar{Y}$
M	$S_{yy} = \sum Y_i^2 - n\bar{Y}^2$
N	個数(データ)n
O	X-MAX
P	X-MIN
Q	Y-MAX
R	Y-MIN
S	回帰係数 a
T	〃 b
U	
V	
W	✓
X	✓
Y	✓
Z	✓
X(9)	入力、推定データ格納
Y(9)	入力、推定データ格納
X1	描線SUB用スタートX
Y1	X Y
X2	〃 エンドX
Y2	〃 Y

PROGRAM  
TITLE

## 借入限度額, ローン返済回数の計算

1

カラーグラフィック  
プリンター(CE-150)必要

## 概要(事務分野)

無理な借金は禁物。

適切な借り入れのための借入限度額の計算と、自分の返済能力から見た返済回数の計算が行なえます。地道な返済プランを……。

## ■使い方

“A” 返済能力(返済額と返済回数)から、借入限度額を計算します。(1円未満切り捨て)

“B” 借入金額と返済能力から返済回数を計算します。

## ■例題

“A” 返済期間8年、返済能力は毎月150万円、年利率12%なら、借入限度額はいくらになるか計算します。

毎年の返済額	= 1500000	} 入力
返済年数	= 8	
利 率	= 12%	

“B” 300万円を借り、年利が12%で返済能力が月10万円であるなら何ヶ月で返済できるか計算します。

借入金額	= 3000000	} 入力
毎月の返済額	= 100000	
利 率	= 12/12%	

## ■内容(計算内容など)

	借入限度額計算	ローン返済回数の計算
入 力	毎回の返済額 (a 円) ローン返済回数 (n 回) 利 率 (r %)	借入金額 (A 円) 毎回の返済額 (a 円) 利 率 (r %)
出 力	借 入 限 度 額	ロ ー ン 返 済 回 数

$$\text{借入限度額} = \frac{a(R^n - 1)}{(R - 1)R^n}$$

$$\left( \text{ただし } R = 1 + \frac{r}{100} \right)$$

$$\text{返済回数} = \frac{\log a - \log(a - A(R - 1))}{\log R}$$

PROGRAM  
TITLE

## 借入限度額, ローン返済回数の計算

2

## ■プリント出力 ※文字の大きさは18字/行

(A)

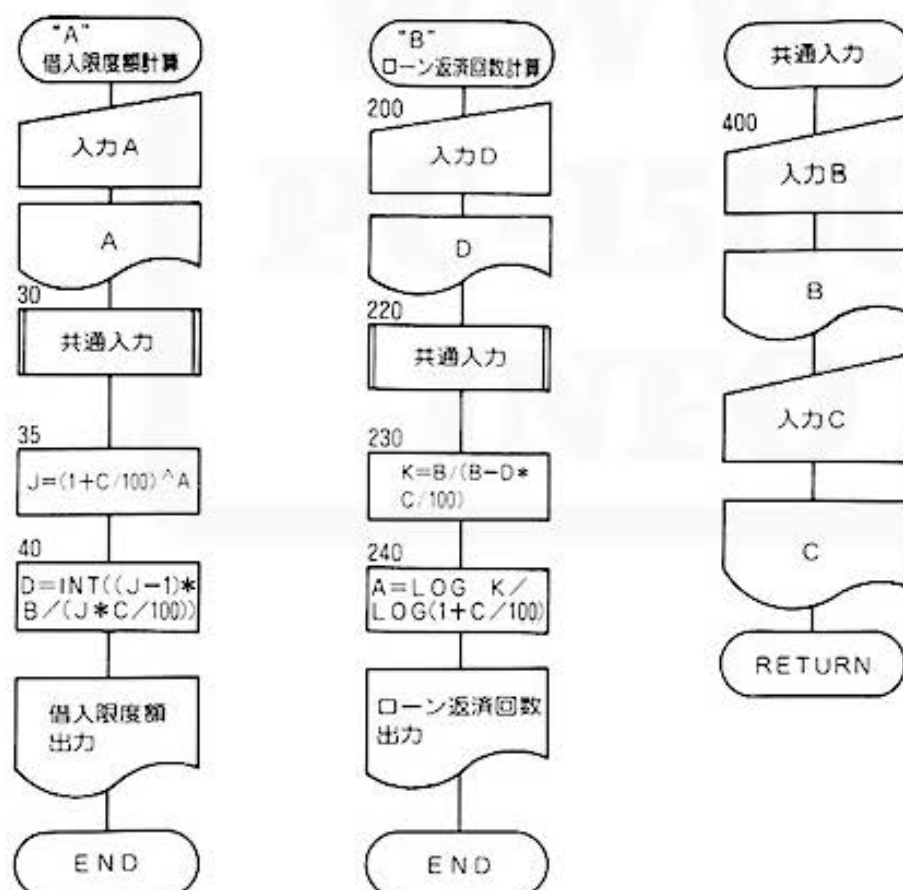
HENSAIKAISU= 8.00  
 HENSAIGAKU=  
           18,000,000  
 RIRITU(%)= 12.000  
 GENDOGAKU=  
           89,417,515

(B)

GENDOGAKU=  
           3,000,000  
 HENSAIGAKU=  
           100,000  
 RIRITU(%)= 1.000  
 HENSAIKAISU= 35.84

## ■フローチャート

A : ローン返済回数  
 B : 毎回の返済額  
 C : 利率(%)  
 D : 借入限度額



PROGRAM  
TITLE

## 借入限度額, ローン返済回数の計算

3

## ■キー操作手順

ステップ	キ ー 入 力	表 示	備 考
1	<b>DEF</b> <b>A</b>	HENSAIKAISU?	
2	8 <b>ENTER</b>	HENSAIGAKU?	
3	1500000 * 12 <b>ENTER</b>	RIRITSU (%) ?	
4	12 <b>ENTER</b>	>	
1	<b>DEF</b> <b>B</b>	GENDOGAKU?	
2	3000000 <b>ENTER</b>	HENSAIGAKU?	
3	100000 <b>ENTER</b>	RIRITSU (%) ?	
4	12 / 12 <b>ENTER</b>	>	

## ■プログラムリスト ※文字の大きさは18字/行

```

10: "A" CLEAR : LF 2
20: INPUT "HENSAIK
   AISU? "; A
25: LPRINT "HENSAI
   KAISU="; USING
   "###.##"; A
30: GOSUB 400
35: J = (1 + C / 100) ^ A
40: D = INT ((J - 1) * B
   / (J * C / 100))
50: LPRINT "GENDOG
   AKU="
55: LPRINT USING "
   #####, #
   ##"; D
60: LF 3: END
200: "B" CLEAR : LF 2
210: INPUT "GENDOGA
   KU? "; D
215: LPRINT "GENDOG
   AKU=": LPRINT
   USING "#####
   ##, ###"; D
220: GOSUB 400
230: K = B / (B - D * C / 100)
240: A = LOG K / LOG (1
   + C / 100)
250: LPRINT "HENSAI
   KAISU="; USING
   "###.##"; A
260: LF 3: END
400: INPUT "HENSAIG
   AKU? "; B
405: LPRINT "HENSAI
   GAKU=": LPRINT
   USING "#####
   #####, ###"; B
410: INPUT "RIRITU(
   %)? "; C
415: LPRINT "RIRITU
   (%)" = "; USING "##
   ###.###"; C
430: RETURN
440: END

```

## メモリー内容

メモリー内容	
A	ローン返済回数
B	毎回の返済額
C	利率 (%)
D	借入限度額
E	
F	
G	
H	
I	
J	計算WORK
K	計算 %
L	
M	
N	
O	
P	
Q	
R	
S	
T	
U	
V	
W	
X	
Y	
Z	

## 概要

高速並びかえ！

業績や順位、名簿などは、手計算で並びかえると非常に時間がかかります。このような煩わしさは、このプログラムで解消し、スピーディーな事務処理をしましょう。

## ■使い方

まず並びかえたいデータが、文字か数値かの選択、そしてデータ数を入力します。次に、小さい順に並びかえるか、又は大きい順に並びかえるかを選択し(文字の場合はA、B、C、…が小さい順で、Z、Y、X、…が大きい順です。)データを入力しますと、並びかえられたデータがプリントされます。

## ■例題

数 値

データ数 7

小さい順

データ	122
	123
	1
	123456
	1234
	11
	12345

文 字

データ数 5

小さい順

データ	CCR
	ABCDE
	RGHI
	KTV
	CCK

## ■内容

入力●文字か数値かの選択

●データ数(標準容量で数値の場合152データまで、文字の場合76データまで)

●小さい順か大きい順の選択

(文字の場合はアスキーコードの大小に従います)

●文字か数値のデータ

出力●並びかえられたデータ

注意：文字データの文字列は16文字までです。



PROGRAM  
TITLE

並びかえ

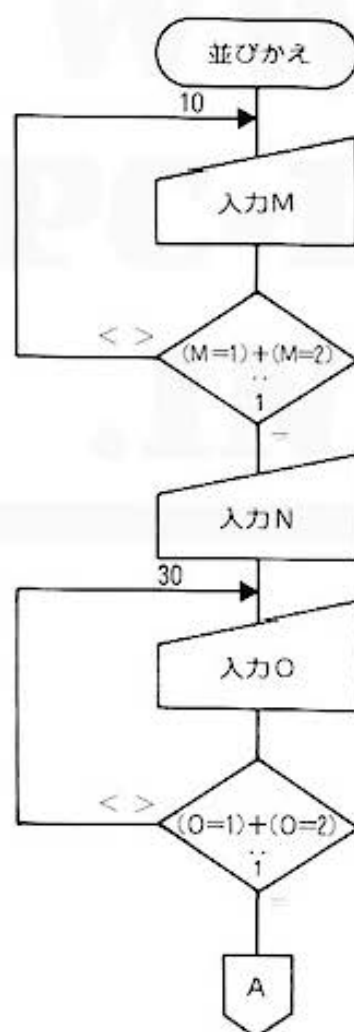
2

## ■プリント出力

1  
11  
122  
123  
1234  
12345  
123456

ABCDE  
CCK  
CCR  
KTU  
RGHI

## ■フローチャート



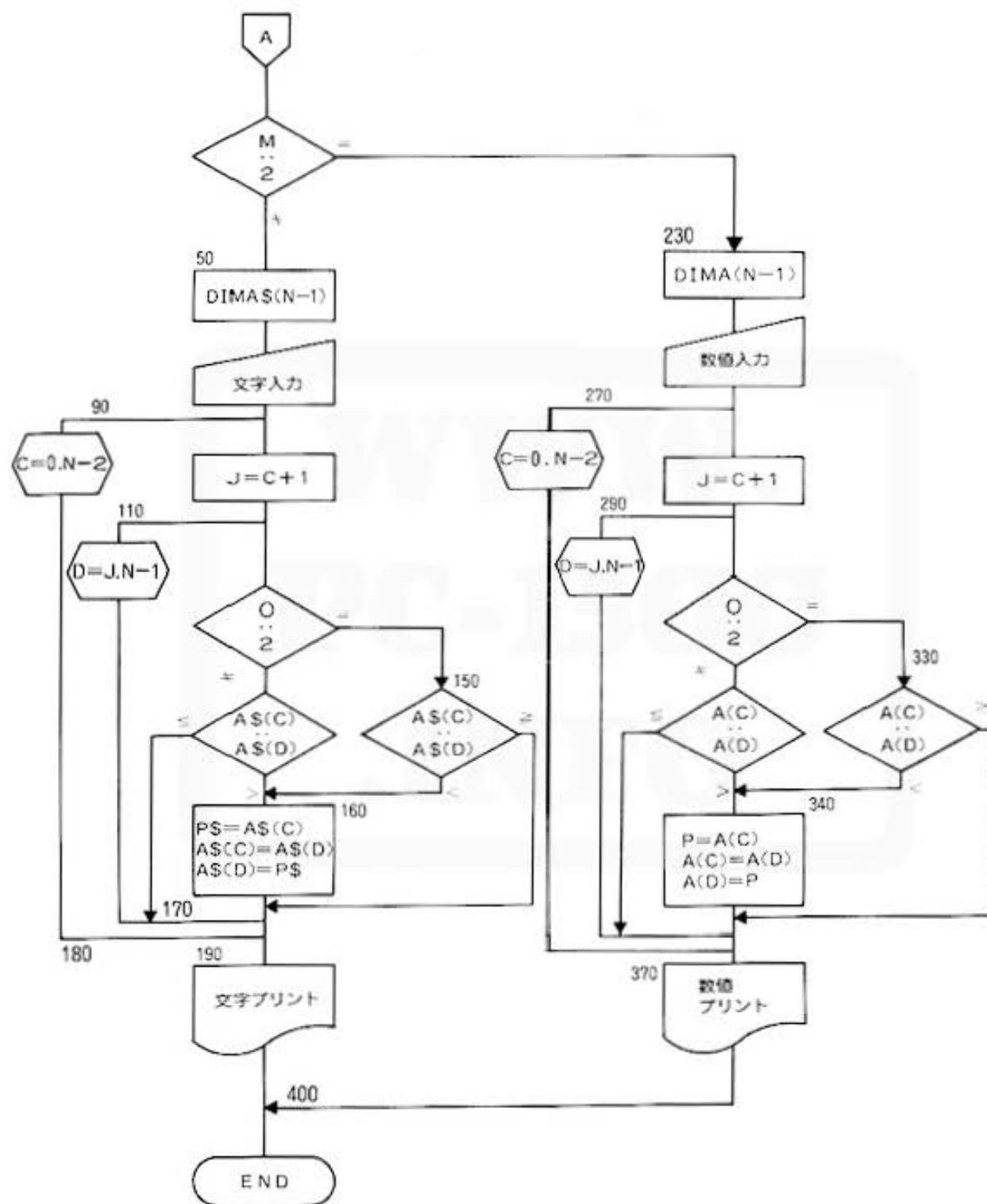
M=1 : 文字  
=2 : 数字  
N : データ数  
O=1 : 昇順  
=2 : 降順

PROGRAM  
TITLE

並び替え

3

## ■フローチャート



PROGRAM  
TITLE

並びかえ

4

## ■キー操作入力

ステップ	人 力 キ ー	表 示	備 考
1	DEF A	MOJI=1 SUUJI=2?	
2	2 ENTER	DATA SUU?	数値の並びかえを選択
3	7 ENTER	CHIISAI=1 OOKII=2?	
4	1 ENTER	X=	
5	122 ENTER	X=	
6	123 ENTER	X=	
7	1 ENTER	X=	
8	123456 ENTER	X=	
9	1234 ENTER	X=	
10	11 ENTER	X=	
11	12345 ENTER	>	プリント出力
<hr/>			
1	DEF A	MOJI=1 SUUJI=2?	
2	1 ENTER	DATA SUU?	文字の並びかえを選択
3	5 ENTER	CHIISAI=1 OOKII=2?	
4	1 ENTER	X\$=	
5	CCR ENTER	X\$=	
6	ABCDE ENTER	X\$=	
7	RGHI ENTER	X\$=	
8	KTV ENTER	X\$=	
9	CCK ENTER	>	プリント出力

PROGRAM  
TITLE

## 並びかえ

5

## ■プログラムリスト ※文字の大きさは18字/行

```

10:"A":CLEAR :
  INPUT "MOJI=1
  SUUJI=2 ?";M
15:IF (M=1)+(M=2)
  <>1GOTO 10
20:INPUT "DATA SU
  ?";N
30:INPUT "CHIISAI
  =1 OKII=2 ?";
  O
35:IF (O=1)+(O=2)
  <>1GOTO 30
40:IF M=2GOTO 230
50:DIM A$(N-1)
60:FOR B=0TO N-1
70:INPUT "X$=";A$(
  B)
80:NEXT B
90:FOR C=0TO N-2
100:J=C+1
110:FOR D=JTO N-1
120:IF O=2GOTO 150
130:IF A$(C)>A$(D)
  GOTO 160
140:GOTO 170
150:IF A$(C)<A$(D)
  GOTO 160
155:GOTO 170
160:P$=A$(C):A$(C)
  =A$(D):A$(D)=P
  $
170:NEXT D
180:NEXT C
190:FOR E=0TO N-1
200:LPRINT A$(E)
210:NEXT E
220:GOTO 400
230:DIM A(N-1)
240:FOR B=0TO N-1
250:INPUT "X=";A(B
  )
260:NEXT B
270:FOR C=0TO N-2
280:J=C+1
290:FOR D=JTO N-1
300:IF O=2GOTO 330
310:IF A(C)<=A(D)
  GOTO 350
320:GOTO 340
330:IF A(C)>A(D)
  GOTO 350
340:P=A(C):A(C)=A(
  D):A(D)=P
350:NEXT D
360:NEXT C
370:FOR E=0TO N-1
380:LPRINT A(E)
390:NEXT E
400:END

```

メモリー内容

A		A\$		A (N)	数値データ
B	ループ処理カウンター	B\$		A\$(N)	文字データ
C	〃	C\$			
D	〃	D\$			
E	〃	E\$			
F		F\$			
G		G\$			
H		H\$			
I		I\$			
J	√	J\$			
K		K\$			
L		L\$			
M	文字、数字の選択コード	MS			
N	データ数	NS			
O	昇順、降順選択コード	OS			
P	並びかえ数字の置換用	PS	並びかえ文字の置換用		
Q		QS			
R		RS			
S		SS			
T		TS			
U		US			
V		VS			
W		WS			
X		XS			
Y		YS			
Z		ZS			

**概要(事務分野)**

得られたデータを、いくつかのランクに分け、それぞれどれぐらいの度数があるか知ることとは、全体像を把握するためにしばしば必要になることです。度数分布図を描きますので、視覚的にとらえることもできます。

**■使い方**

1. パラメータの入力(データ数、級の初期値、級の幅、級の数)
  2. データの入力方法設定[キー入力又はカセット入力]  
 キー入力 : データをキーから入力し、その後カセット・テープへデータ出力  
 カセット入力: データをカセットテープから入力します。
  3. 分散及び標準偏差を計算し、プリンターへ出力します。
  4. 度数分布図をプリンターへ出力します。
- 注) データは最大121までとします。

**■例題**

データ数=10、級の初期値=0、級の幅=2、級の数=5

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

分散 : 6.81

標準偏差: 2.60959767

**■内容(計算内容など)**

$$V = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2 \quad (\text{分散})$$

$$S = \sqrt{V} \quad (\text{標準偏差})$$

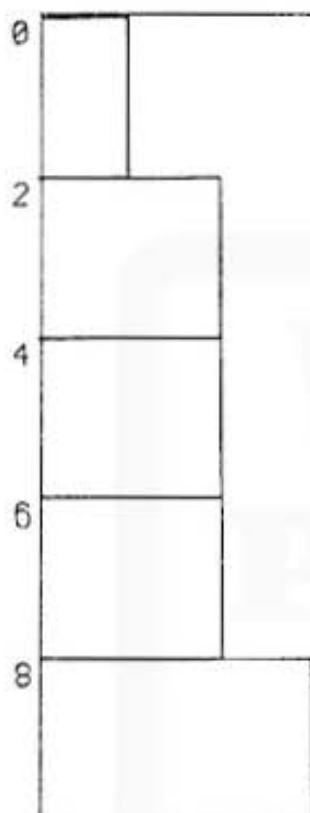
PROGRAM  
TITLE

## 度数分布図

2

## ■プリント出力

※文字の大きさは18字/行

(実際はカラープリント出力です。)  
(P.121参照)BUNSAN= 6.81  
HYOJUN HENSA=  
2.60959767

## ■プログラムリスト

※文字の大きさは18字/行

```

10:"A":CLEAR:
  TEXT:USING
20:INPUT "DATA-SU
  =" ;N
30:INPUT "SHOKI-C
  HI=" ;F
40:INPUT "KUKAN-H
  ABA=" ;B
50:INPUT "KUKAN-S
  U=" ;M
60:DIM A1(N-1),H(
  M-1)
70:FOR C=0TO M-1
80:H(C)=0
90:NEXT C
100:INPUT "KEY-IN?
  (Y,N)";A$
110:IF A$="N"GOTO
  170
115:Z=F+B*M-1:X=0
120:FOR C=0TO N-1
130:INPUT "DATA=";
  A1(C):GOTO 150
140:GOTO 160
150:IF A1(C)>ZGOTO
  130
152:IF A1(C)<FGOTO
  130
153:X=X+1
155:NEXT C
160:PRINT #X,A1(*)
165:GOTO 180
170:INPUT #X,A1(*)
180:S=0:N=X
190:FOR C=0TO N-1
200:I=INT ((A1(C)-
  F)/B)
210:H(I)=H(I)+1
220:S=S+A1(C)
230:NEXT C
240:U=S/N:T=0
250:FOR C=0TO N-1
260:T=T+(A1(C)-U)^
  2
270:NEXT C
280:T=T/N:S=J T
290:COLOR 0:LPRINT
  "BUNSAN=" ;T
300:LPRINT "HYOJUN
  HENSA=" ;S
310:N=-10^(98)
320:FOR C=0TO M-1
330:IF H(C)>NLET N
  =H(C)
340:NEXT C
350:GRAPH
360:GLCURSOR (50,0
  ):SORGN
370:COLOR 0
380:LINE (0,0)-(15
  0,0)
390:LINE (0,0)-(0,
  -450)
400:L=450/M:N=N/15
  0
410:W=0:Q=F
420:FOR C=0TO M-1
422:COLOR 2:
  GLCURSOR (-50,
  W-15)
424:LPRINT USING "
  ####";Q
430:COLOR 1
435:G=INT (H(C)/N)
440:LINE (0,W)-(G,
  W)-(G,W-L)-(0,
  W-L)
450:W=W-L
470:Q=Q+B
480:NEXT C
490:END

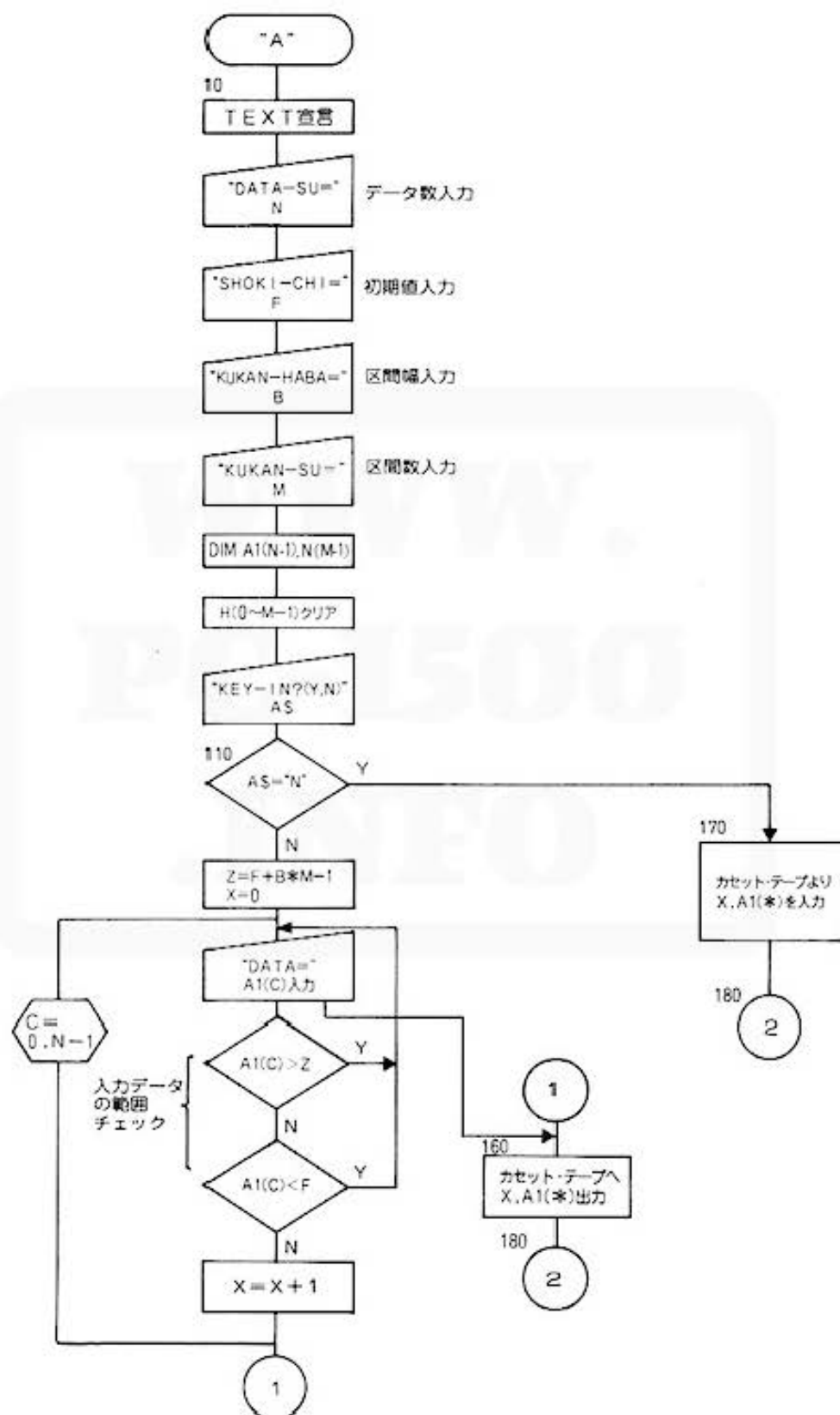
```

PROGRAM  
TITLE

度数分布図

3

## ■フローチャート



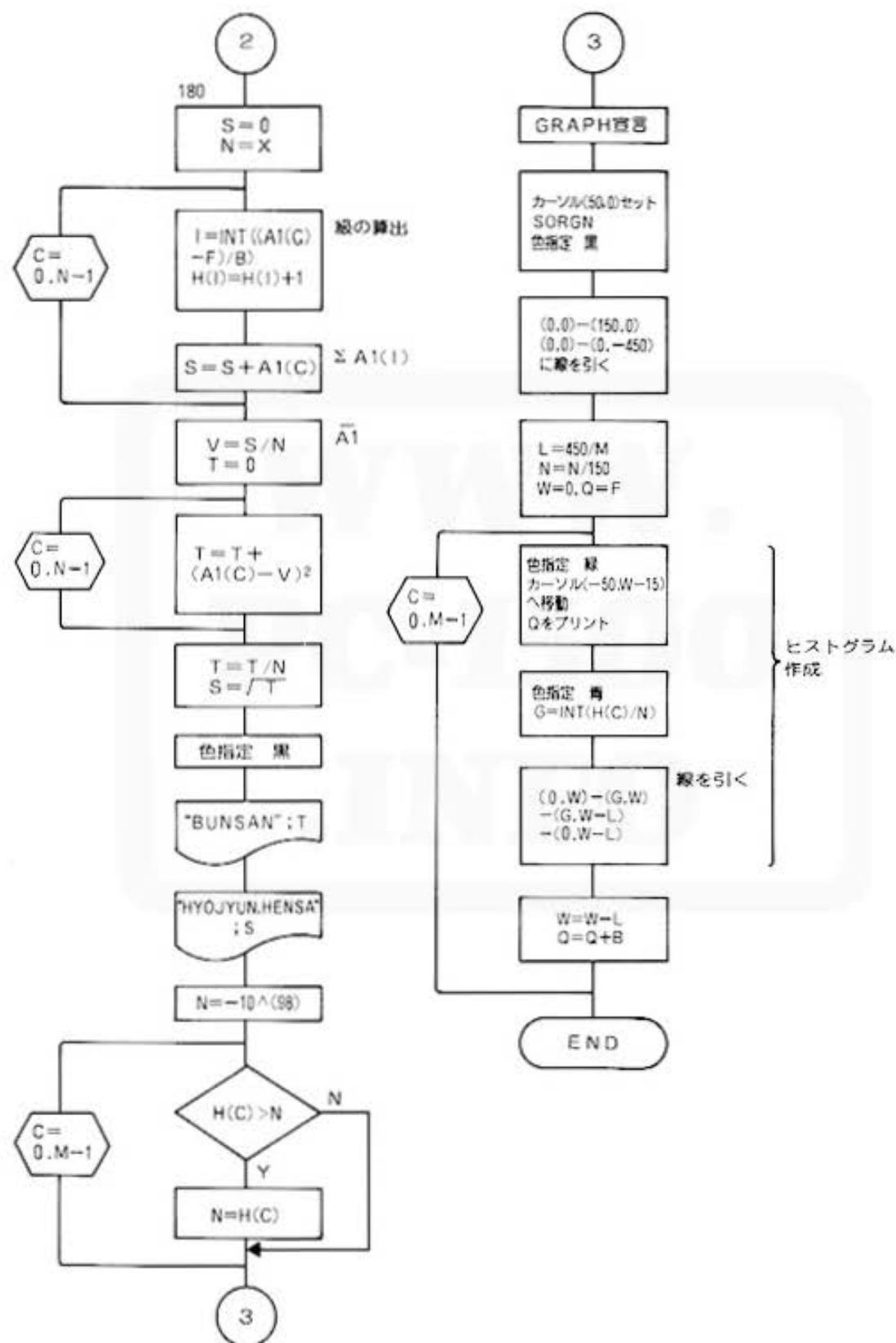


PROGRAM  
TITLE

## 度数分布図

3

## ■フローチャート



PROGRAM  
TITLE

## 度数分布図

4

## ■キー操作手順

ステップ	キー入力	表示	備考
1	<b>DEF</b> <b>A</b>	DATA-SU=	
2	10 <b>ENTER</b>	SHOKI-CHI=	
3	0 <b>ENTER</b>	KUKAN-HABA=	
4	2 <b>ENTER</b>	KUKAN-SU=	
5	5 <b>ENTER</b>	KEY-IN?(Y, N)	Y: キーボードから N: カセットテープから
6	Y <b>ENTER</b>	DATA=	
7	5 <b>ENTER</b>	DATA=	
8	2 <b>ENTER</b>	DATA=	
9	⋮	⋮	

メモリー内容				
A		AS	✓	A1(N-1) データ・テーブル
B	区間幅	BS		
C	✓	CS		H(M-1) 級データ・テーブル
F	初期値	FS		
G	✓	GS		
I	級NO	IS		
L	✓	LS		
M	区間数	MS		
N	データ数	NS		
Q	✓	QS		
S	$\sum A1(i), \sqrt{T}$	SS		
T	$\frac{1}{N} \sum (A1(i) - \bar{A})^2$	TS		
V	平均値	VS		
W	✓	WS		
X	有効データ数	XS		
Z	有効最大値	ZS		

**概要**

今月のあなたの状態は？

心と体の状態は健康のバロメーター。その日のあなたの運勢をも左右する大きな原因になります。そこで未然に自分のバイオリズムがわかれば、とても便利ですね。あなたの名前と生年月日を入れると、あなたの希望年月の(1ヶ月分)バイオリズムグラフがプリントされ、身体(緑色)、感情(赤色)、知性(青色)の曲線があなたの運勢にアドバイスを与えてくれます。

**■使い方**

- まず希望する年月を入力します。次に氏名(16文字まで)と生年月日を入力します。(全て西暦で入力して下さい。)
- 希望年月のバイオリズムが、各要素毎に色を変えてプリントされます。

**■例題**

希望年月	1981年7月	} を入力します。
氏 名	SHARP	
生年月日	1952年1月28日	

**■内容**

入力：調査年月

名前

生年月日

出力：調査年月のバイオリズム曲線を色を変えてプリントします。

曲線の計算式

身体  $X = \sin((B+Y)/23 \times 360) \times 80$

感情  $X = \sin((C+Y)/28 \times 360) \times 80$

知性  $X = \sin((D+Y)/33 \times 360) \times 80$

B、C、D：生年月日から調査年月までの通算日数を各周期で割った余りの日数

Y：日付

(+)、(-)方向に各々16mmをMAXとしています。

周期 身体(23日)、感情(28日)、知性(33日)

PROGRAM  
TITLE

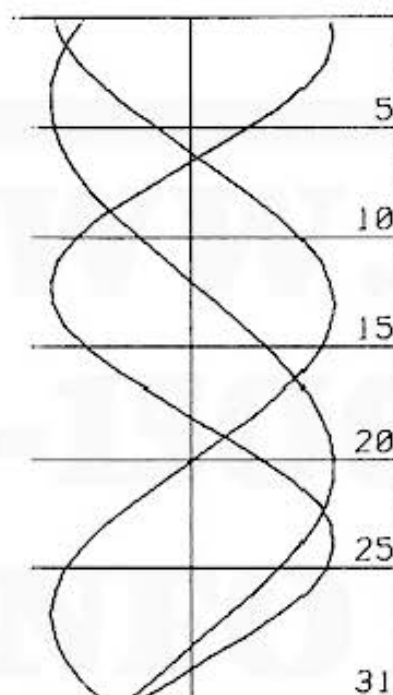
バイオリズム

2

## ■プリント出力 ※文字の大きさは18字/行

(実際はカラープリント出力です。)  
(P.121参照)DATE 1981, 7  
NAME SHARP  
BIRTH 1952, 1, 28-- SHINTAI  
-- KANJYO  
-- CHISEI

(-) (+)



## ■キー操作手順

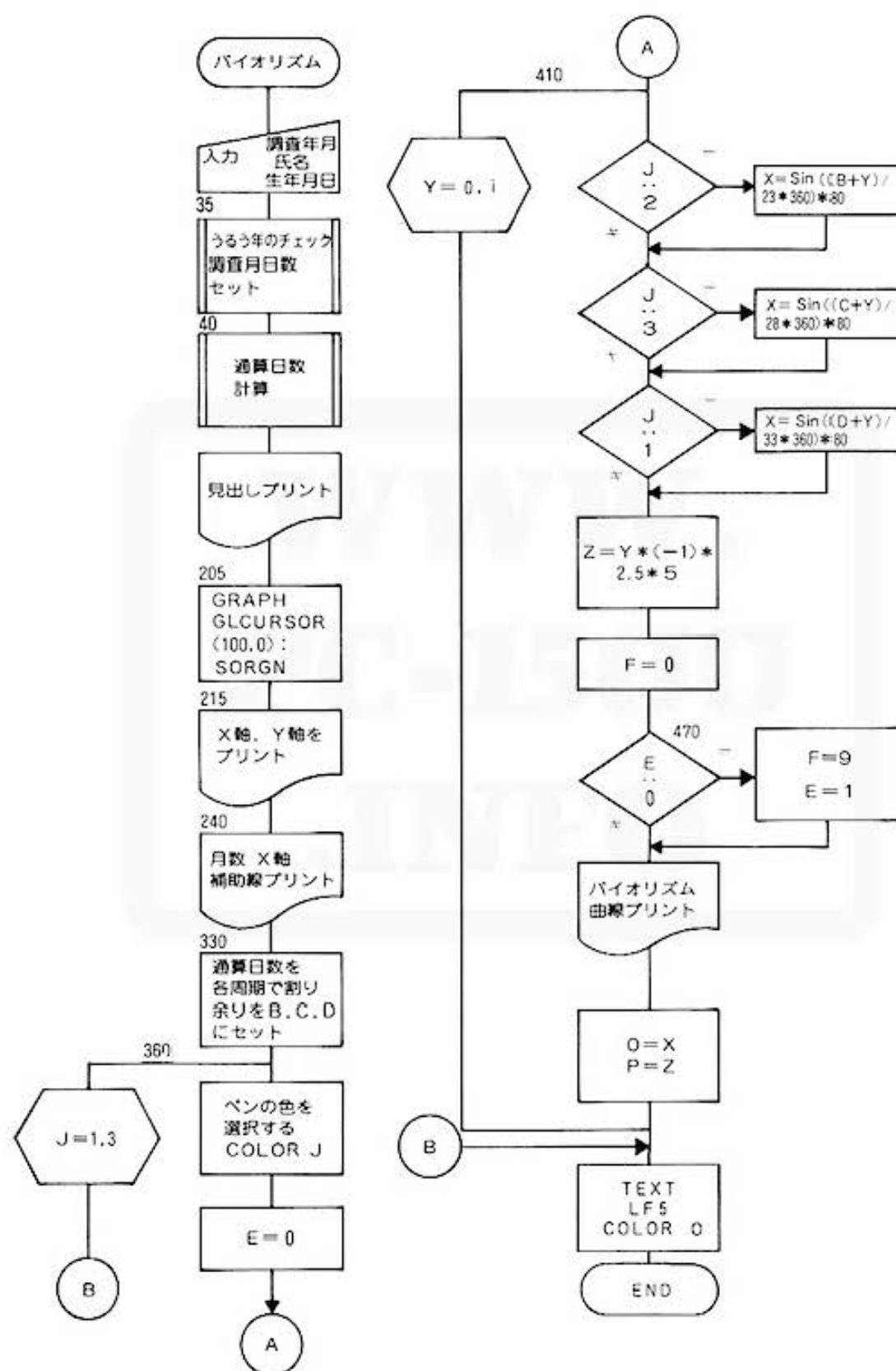
ステップ	キー入力	表示	備考
1	DEF A	DATE?YEAR=	
2	1981 ENTER	MONTH=	西暦で4桁入力
3	7 ENTER	NAME?	
4	SHARP ENTER	BIRTH?YEAR=	
5	1952 ENTER	MONTH=	西暦で4桁入力
6	1 ENTER	DAY=	
7	28 ENTER	>	バイオリズム曲線出力

PROGRAM  
TITLE

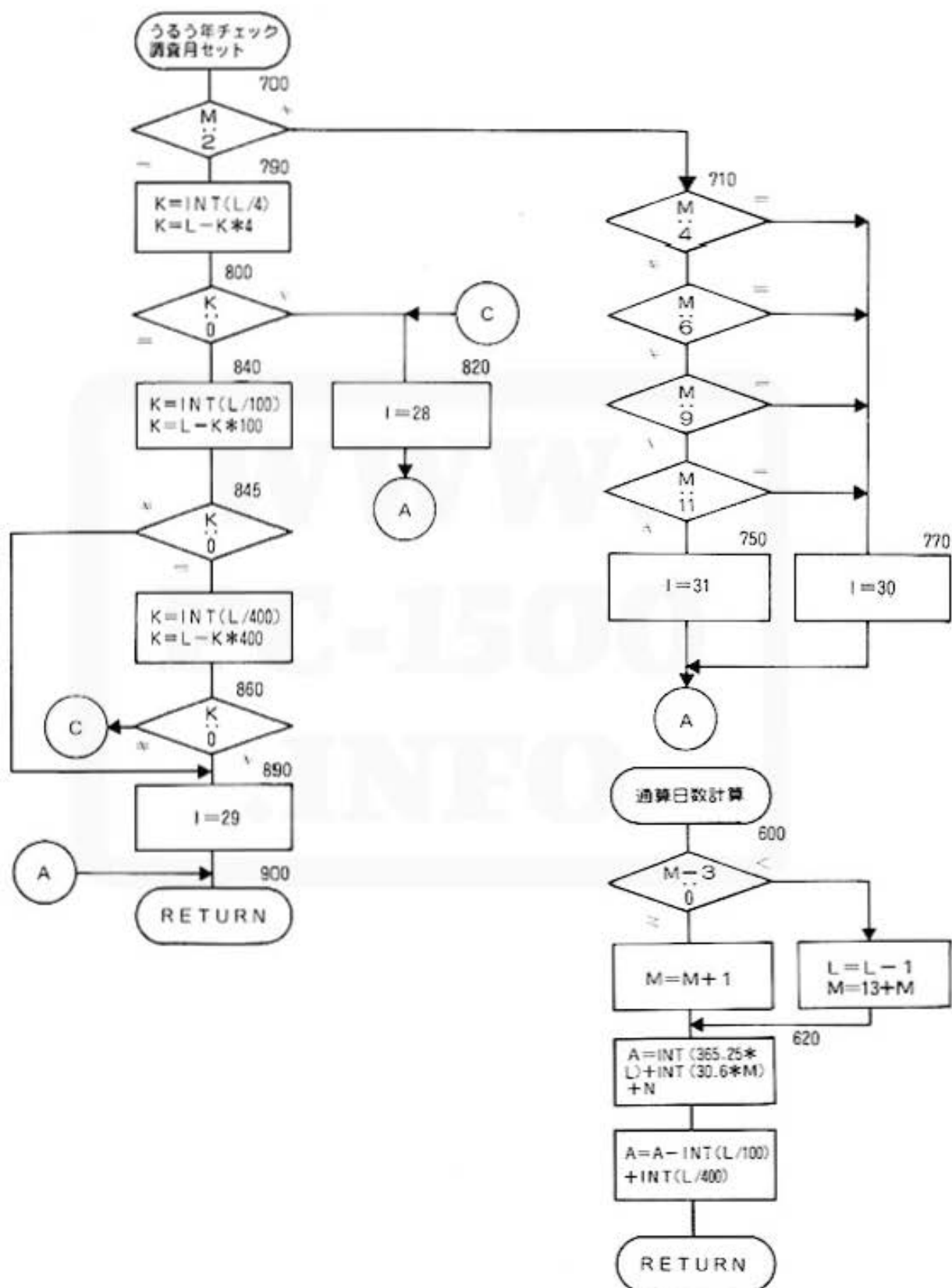
バイオリズム

3

## ■フローチャート



## ■フローチャート



PROGRAM  
TITLE

## バイオリズム

4

## ■プログラムリスト ※文字の大きさは18字/行

```

10:"A":CLEAR:
  INPUT "DATE?
  YEAR=";L,"MONT
  H=";M
15:TEXT:COLOR 0
20:LPRINT "DATE";
  USING "#####";
  L;";";USING "H
  ##";M
30:N=0
35:GOSUB 700
40:GOSUB 600:O=A
50:INPUT "NAME? "
  ;A$
60:LPRINT "NAME "
  ;A$
70:INPUT "BIRTH?
  YEAR=";L,"MONT
  H=";M,"DAY=";N
80:LPRINT "BIRTH"
  ;USING "#####";
  L;";";USING "
  ###";M;";";
  USING "###";N
90:GOSUB 600:P=A
100:A=0-P:O=0:P=0
110:LF 2
120:COLOR 2
130:LPRINT " -- SH
  INTA!"
140:COLOR 3
150:LPRINT " -- KA
  NJYO"
160:COLOR 1
170:LPRINT " -- CH
  ISEI"
180:LF 1
190:COLOR 0
200:LPRINT " (-)
  (+)"
205:GRAPH
210:GLCURSOR (100,
  0):SORGN
215:Y=1*2.5*5*(-1)
220:LINE (-100,0)-
  (115,0)
230:LINE (0,0)-(0,
  Y)
235:LINE (115,Y)-(
  115,0)
240:FOR Q=5TO 30
  STEP 5
243:R=Q
245:IF Q=30LET R=1
250:Y=R*2.5*(-1)*5
260:LINE (-90,Y)-(
  115,Y)
270:X=80
290:Z=Y+5
300:LINE (115,Z)-(
  X,Z),9
310:LPRINT R
320:NEXT Q
330:B=INT (A/23):B
  =A-(23*B)
340:C=INT (A/28):C
  =A-(28*C)
350:D=INT (A/33):D
  =A-(33*D)
360:FOR J=1TO 3
395:COLOR J
400:E=0
410:FOR Y=0TO 1
420:IF J=2LET X=
  SIN ((B+Y)/23*
  360)*80
430:IF J=3LET X=
  SIN ((C+Y)/28*
  360)*80
440:IF J=1LET X=
  SIN ((D+Y)/33*
  360)*80
450:Z=Y*(-1)*2.5*5
460:F=0
470:IF E=0LET F=9:
  LET E=1
480:LINE (0,P)-(X,
  Z),F
490:O=X:P=Z
500:NEXT Y
510:NEXT J
515:TEXT:LF 5:
  COLOR 0
520:END
600:IF M-3>=0LET M
  =M+1:GOTO 620
610:L=L-1:M=13+M
620:A=INT (365.25*
  L)+INT (30.6*M
  )+N
625:A=A-INT (L/100
  )+INT (L/400)
630:RETURN
640:END
700:IF M=2GOTO 790
710:IF M=4GOTO 770
720:IF M=6GOTO 770
730:IF M=9GOTO 770
740:IF M=11GOTO 77
  0
750:I=31:GOTO 900
770:I=30:GOTO 900
790:K=INT (L/4):K=
  L-K*4
800:IF K=0GOTO 840
820:I=28:GOTO 900
840:K=INT (L/100):
  K=L-K*100
845:IF K=0GOTO 850
847:GOTO 890
850:K=INT (L/400):
  K=L-K*400
860:IF K=0GOTO 890
870:GOTO 820
890:I=29
900:RETURN
910:END

```

## メモリー内容

A	生年月日から調査年月日までの 通算日数
B	通算日数を周期で割った余りを セット (身体)
C	〃 (感情)
D	〃 (知性)
E	1回目のジャッジ
F	PEN-UP, DOWNのコード
G	
H	
I	該当月 日数
J	ループカウンタ
K	うるう年計算 WORK
L	生年 調査年
M	生月 調査月
N	生日
O	西暦0年から調査年月日までの日数
P	西暦0年から生年月日までの日数
Q	ループカウンタ
R	見出し用日数WORK
S	
T	
U	
V	
W	
X	バイオリズム曲線X座標
Y	1ヵ月の日数カウンタ
Z	バイオリズム曲線Y座標
A\$	氏名



**概要**

クイックキーオペレーション！

君はキーボードをどれだけ速く、正確に押せるか。

このプログラムで練習して、キーオペレーションをぐっと楽にしておけばプログラムの打ち込みも迅速処理。自己の可能性を高めよう。

**■使い方(遊び方)**

ブザーが鳴ってから3～6文字のアルファベットが表示されますから、それと同じものを制限時間内にキーボードから入れます。

全て合っていれば10点、半数以上合っていれば5点が合計得点に加算されます。制限時間を越えると次の問題が出て来ます。制限時間はグレード(1、2、3)と、表示される文字数により決まりますが、グレードの1が最も制限時間が短く、3が最も長くなっています。

同じグレード内で得点を競います。

問題は10問、最高得点は100点です。

**■内容(計算内容など)**

文字数(3～6)は乱数により決めます。

文字列(A～Z)も乱数により抽出します。

時間はグレードと文字数によって決まります。

PROGRAM  
TITLE

タイプ練習

2

## ■プログラムリスト

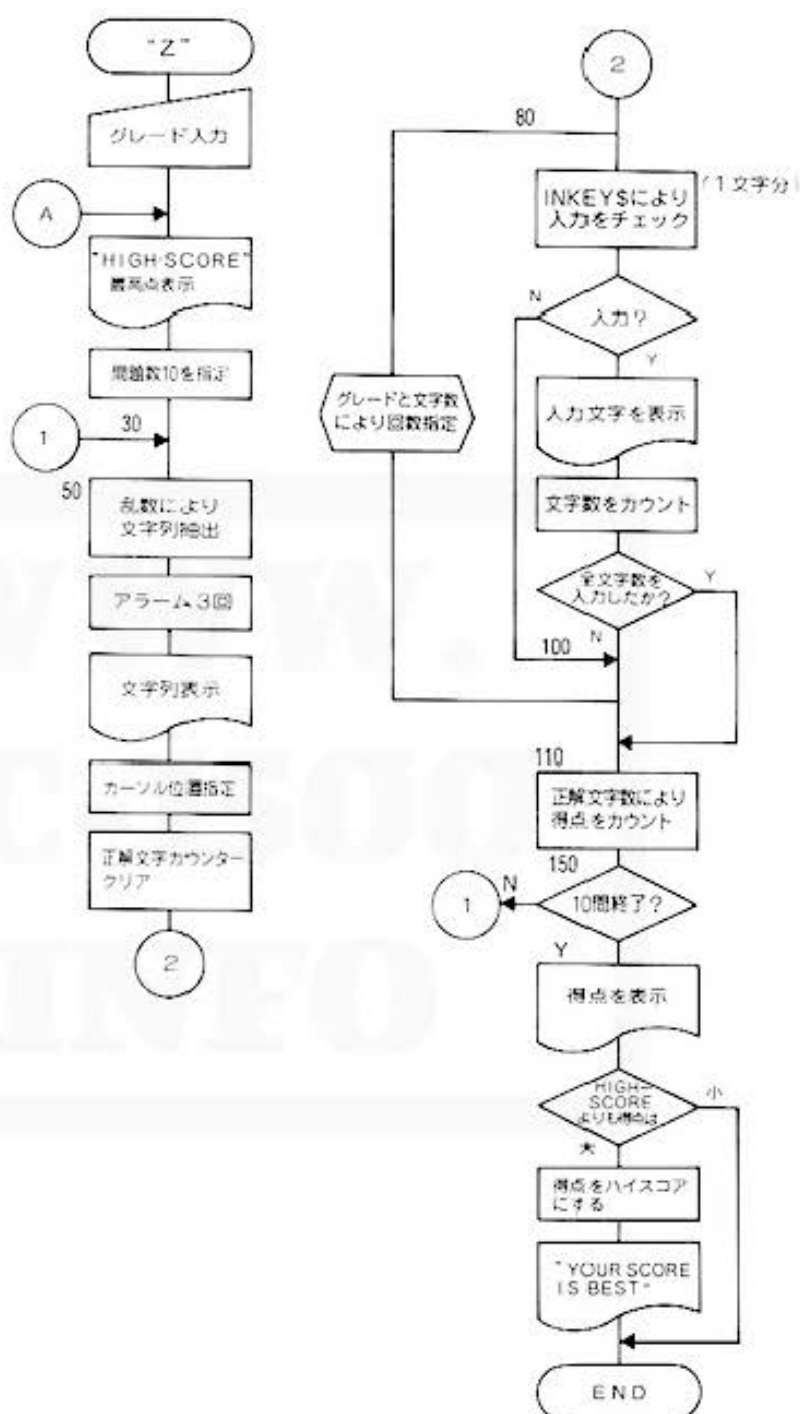
※文字の大きさは18字/行

```

10:"Z":CLEAR:CLS
  :DIM A$(5),B$(
  5):RANDOM
15:INPUT "GRADE(1
  ,2,3)?":L
17:IF (L=1)+(L=2)
  +(L=3)<>1THEN
  15
20:"A":WAIT 0:P=0
  :PAUSE "HIGH-S
  CORE=":X
30:FOR S=1TO 10
40:B=RND 4+2:Y$="
  ":R=INT (B/2)
50:FOR C=0TO B-1:
  B$(C)=" "
60:D=RND 26:A$(C)
  =CHR$(D+&40):
  Y$=Y$+CHR$(D+
  &40):NEXT C
70:CLS:BEEP 3:
  PRINT Y$:
  CURSOR 10:E=0
80:FOR W=1TO B*10
  *L:B$(E)=
  INKEY$:IF B$(
  E)=""THEN 100
85:PRINT B$(E);
90:E=E+1:IF E=B
  LET W=400
100:NEXT W:Q=0
110:FOR W=0TO B-1:
  IF A$(W)=B$(W)
  LET Q=Q+1
120:NEXT W:IF Q<=R
  THEN 150
130:IF Q=BLET P=P+
  10:GOTO 150
140:P=P+5
150:NEXT S:CLS:
  BEEP 3:PAUSE "
  YOUR-SCORE=":P
160:IF P>XLET X=P:
  PRINT "YOUR SC
  ORE IS BEST"
170:WAIT:PRINT:
  END

```

## ■フローチャート



PROGRAM  
TITLE

## タイプ練習

3

## ■キー操作手順

ステップ	キー入力	表示	備考
1	<b>DEF</b> <b>Z</b>	GRADE(1, 2, 3)?	グレードの変更又は最初にこのゲームを行う時1, 2, 3のいずれかを <b>ENTER</b> する。
2	<b>DEF</b> <b>A</b>	HIGH-SCORE=XXX	<b>DEF</b> <b>Z</b> を入力の際は必要なし。
3		XXXXXX (6文字以内の文字列)	アラーム3回
4	++++++ ステップ3で表示された文字列	XXXXXX ++++++ 入力された文字	全文字を入力するか、一定時間過ぎると問題がかわる。10回くり返す。
:	:	:	
		YOUR-SCORE=XXX	アラーム3回 得点が表示されます。
		YOUR SCORE IS BEST	最高点を超える得点の場合にのみ表示されます。

メモリー内容				
A		A\$		A\$(0~5) 乱数により発生させた文字
B	文字数(3~6)	B\$		B\$(0~5) 入力された文字
C	✓	C\$		
D	✓	D\$		
E	入力文字数	E\$		
F		F\$		
G		G\$		
H		H\$		
I		I\$		
J		J\$		
K		K\$		
L	グレード	L\$		
M		M\$		
N		N\$		
O		O\$		
P	得点	P\$		
Q	正解文字数	Q\$		
R	✓	R\$		
S	問題数	S\$		
T		T\$		
U		U\$		
V		V\$		
W	時間	W\$		
X	最高得点	X\$		
Y		Y\$	問題の文字列	
Z		Z\$		

## 概要

このプログラムではPC-1500に内蔵の時計は、こんなにも便利だという例をお見せします。これ一台で、ストップウォッチもタイマーも目覚し時計もOKです。

## ■使い方(遊び方)

(A)アラーム時刻(0~23時0~59分0~59秒)を入力しますと、予定の時刻にメロディで知らせてくれます。

この間、時刻も表示します。

(D)タイマー時間をセットして **SPACE** キーでスタートを入力すると指定時間経過後、メロディで知らせてくれます。

この間、経過時間も表示されます。

(S)ストップウォッチと同様に **SPACE** キーでスタートとエンドを入力します。この間、経過時間が表示されます。

(注意)このプログラムを使用する前に、内蔵時計(TIME)を正しくあわせて下さい。

## ■例題

(A)アラーム時刻10時57分00秒としますと、105700と入力します。

**NOW-TIME 10H 54M 5S** と現在時刻が表示されつづけ10時57分00秒になりますとメロディがなります。

(D)タイマー時間を0時間30分00秒としますと、003000と入力します。

**SPACE** を押しますと、**KEIKA 0H 0M 1S** のように経過時間が表示されつづけ、0時間30分00秒が経過すると、メロディがなります。

(S) **SPACE** を押しますと **0M 0S 2SS** のイメージで経過時間が流れます。次に **SPACE** を押しますと、**STOP 10M59S 4SS** 経過時間が表示され、処理は終了します。

## ■内容(計算内容など)

(A)目覚し時計の働き

アラーム時刻(時、分、秒)を入力します。

予定時間になればメロディを発生して知らせます。この間、その時の時刻を表示します。

(D)タイマーの働き

タイマー時間(時間、分、秒)を入力します。

**SPACE** でスタートして指定時間経過後、メロディを発生して知らせます。この間秒単位で経過時間を表示します。

(S)ストップウォッチの働き

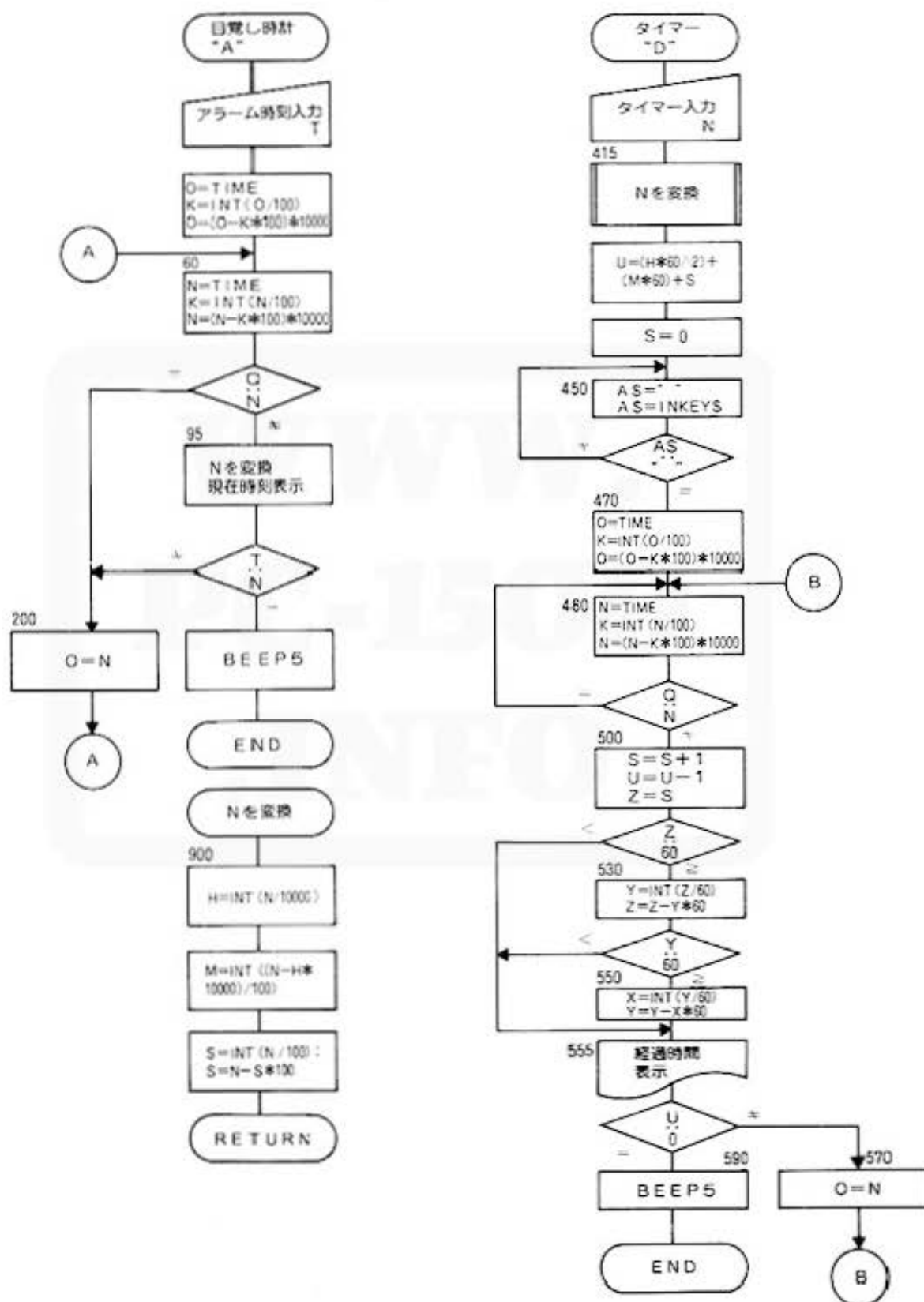
**SPACE** でスタートして **SPACE** で終了します。この間1/5秒単位で経過時間を表示します。

PROGRAM  
TITLE

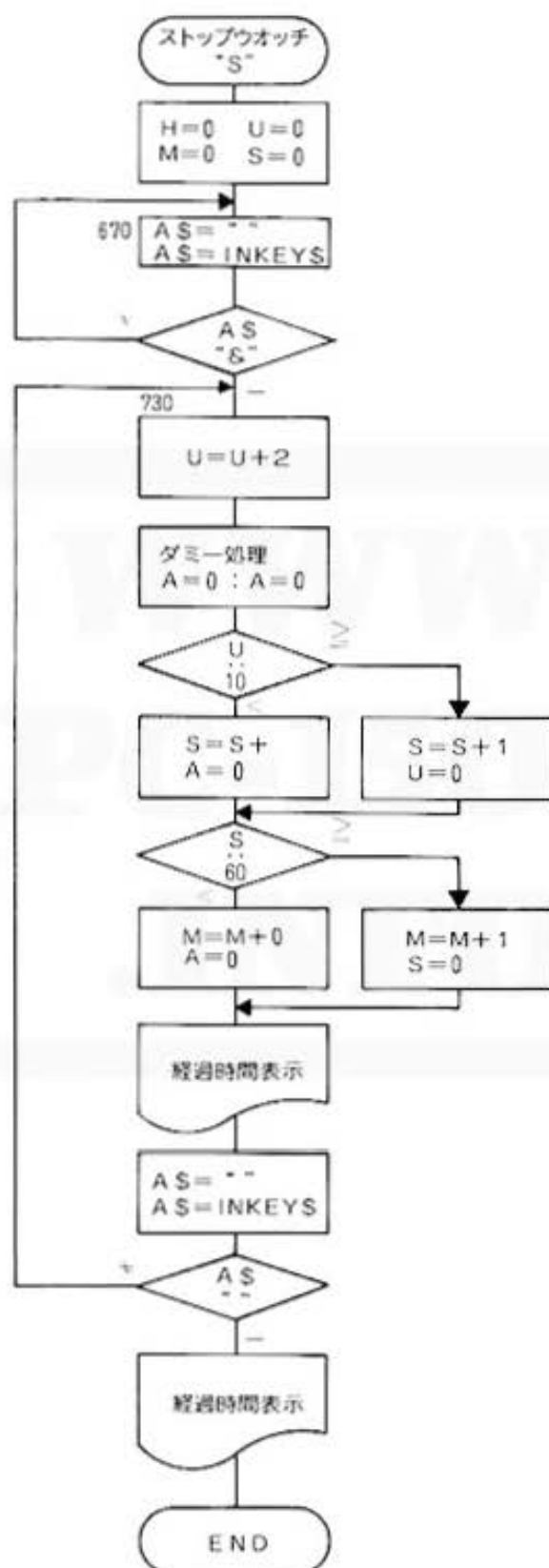
ストップウォッチ, タイマー, 目覚し時計

2

## ■フローチャート



## ■フローチャート



PROGRAM  
TITLE

ストップウォッチ, タイマー, 目覚し時計

3

## ■キー操作手順

ステップ	キー入力	表示	備考
1	TIME=10110.5350 ENTER	10110.535	現時刻を入力 1月1日10時53分50秒
2	DEF A	ALARM-TIME?	アラームタイム指定 10時57分00秒
3	105700 ENTER	NOW-TIME 10H53M56S	現時刻表示
		.....	
4		NOW-TIME 10H57M00S	アラームが20回鳴る
1	DEF D	TIMER?	時周指定 00時30分00秒後セット
2	003000 ENTER	TIMER003000	
3	SPACE	KEIKA 0H 0M 1S	
		.....	
4		KEIKA 0H 30M 0S	アラームが20回鳴る
1	DEF S	STOP WATCH	
2	SPACE	0M 0S 0SS	ストップウォッチスタート
		.....	
3	SPACE	STOP 1M 0S 0SS	1分後



PROGRAM  
TITLE

## ストップウォッチ, タイマー, 目覚し時計

4

## ■プログラムリスト ※文字の大きさは18字/行

```

10: "A":CLEAR :
   WAIT 0
20: INPUT "ALARM-T
   IME?";T
30: O=TIME
40: K=INT (O/100)
50: O=(O-K*100)*10
   000
60: N=TIME
70: K=INT (N/100)
80: N=(N-K*100)*10
   000
90: IF O=NGOTO 200
95: GOSUB 900
100: PRINT "NOW-TIM
   E";USING "###"
   ;H;"H";USING "
   ###";M;"M";
   USING "###";S;
   "S"
120: IF T=NBEEP 20:
   GOTO 300
200: O=N
210: GOTO 60
300: END
400: "D":CLEAR :
   WAIT 0
410: INPUT "TIMER?"
   ;N
415: GOSUB 900
420: U=(H*60^2)+(M*
   60)+S
440: S=0
445: A$=""
450: A$=INKEY$
460: IF A$<>" "GOTO
   445
470: O=TIME :K=INT
   (O/100):O=(O-K
   *100)*10000
480: N=TIME :K=INT
   (N/100):N=(N-K
   *100)*10000
490: IF O=NGOTO 480

500: S=S+1
505: U=U-1
510: Z=S
520: IF Z<60GOTO 55
   S
530: Y=INT (Z/60):Z
   =Z-Y*60
540: IF Y<60GOTO 55
   S
550: X=INT (Y/60):Y
   =Y-X*60
555: PRINT "KEIKA";
   USING "##";X;"
   H";USING "###"
   ;Y;"M";USING "
   ###";Z;"S"
560: IF U=0GOTO 590
570: O=N:GOTO 480
590: BEEP 20
600: END
650: "S":CLEAR :
   WAIT 0
655: PRINT "STOP WA
   TCH"
660: H=0:M=0:S=0:U=
   0
670: A$=""
680: A$=INKEY$
690: IF A$<>" "GOTO
   670
730: U=U+2
735: A=0:A=0
740: IF U<10LET S=S
   +0:A=0
750: IF U=10LET S=S
   +1:U=0
760: IF S<60LET M=M
   +0:A=0
770: IF S=60LET M=M
   +1:S=0
810: PRINT M;"M";
   USING "###";S;
   "S";USING "##"
   ;U;"SS"

815: A$=""
820: A$=INKEY$
830: IF A$<>" "GOTO
   730
840: WAIT :USING :
   PRINT "STOP";M
   ;"M";S;"S";U;"
   SS"
850: END
900: H=INT (N/10000
   )
910: M=INT ((N-H*10
   000)/100)
920: S=INT (N/100):
   S=N-S*100
930: RETURN
940: END

```

PROGRAM  
TITLE

## ストップウォッチ, タイマー, 目覚し時計

5

メモリー内容(ストップウォッチ)		メモリー内容(タイマー)		メモリー内容(目覚し時計)	
A	WORK	A		A	
B		B		B	
C		C		C	
D		D		D	
E		E		E	
F		F		F	
G		G		G	
H	経過時間(時間)	H	タイマー時間(時)	H	現在時刻(時)
I		I		I	
J		J		J	
K		K	計算WORK	K	計算WORK
L		L		L	
M	経過時間(分)	M	タイマー時間(分)	M	現在時刻(分)
N		N	タイマー時間:経過時間(NOW)	N	経過時間(NOW)
O		O	経過時間(OLD)	O	経過時間(OLD)
P		P		P	
Q		Q		Q	
R		R		R	
S	経過時間(秒)	S	タイマー時間(秒)	S	現在時刻(秒)
T		T		T	アラーム時間
U	経過時間(1/10秒)	U	タイマー時間を秒と換算	U	
V	経過時間	V		V	
W		W		W	
X		X	経過時間(時)	X	
Y		Y	経過時間(分)	Y	
Z		Z	経過時間(秒)	Z	
AS	INKEY\$	AS	INKEY\$		

カラーグラフィック  
プリンター(CE-150)必要**概要**

カラーグラフィックプリンタを用いて、いろいろなデザインが作れるようになれば非常に楽しいですね。それでは、このプログラムで、花の図柄を描いて見ましょう。

**■使い方(遊び方)**

**DEF** **A** を押せば、かわいい花が描き出されます。

**■内容(計算内容など)**

$$X(i) = \sin(6 * i) * \cos(i + A) * 80$$

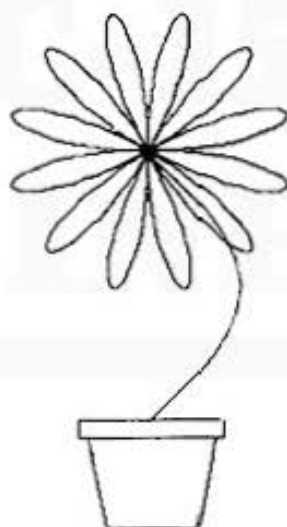
$$Y(i) = \sin(6 * i) * \sin(i + A) * 80$$

1つの花びらに対して上記iの値を1～30まで変化させ30座標を線分で結びます。

Aの値を0°～330°まで30°づつ12回変化させることにより、12枚の花びらの変化した座標を求めます。

**■プリント出力**

(実際はカラープリント出力です。)  
(P.121参照)

**■キー操作手順**

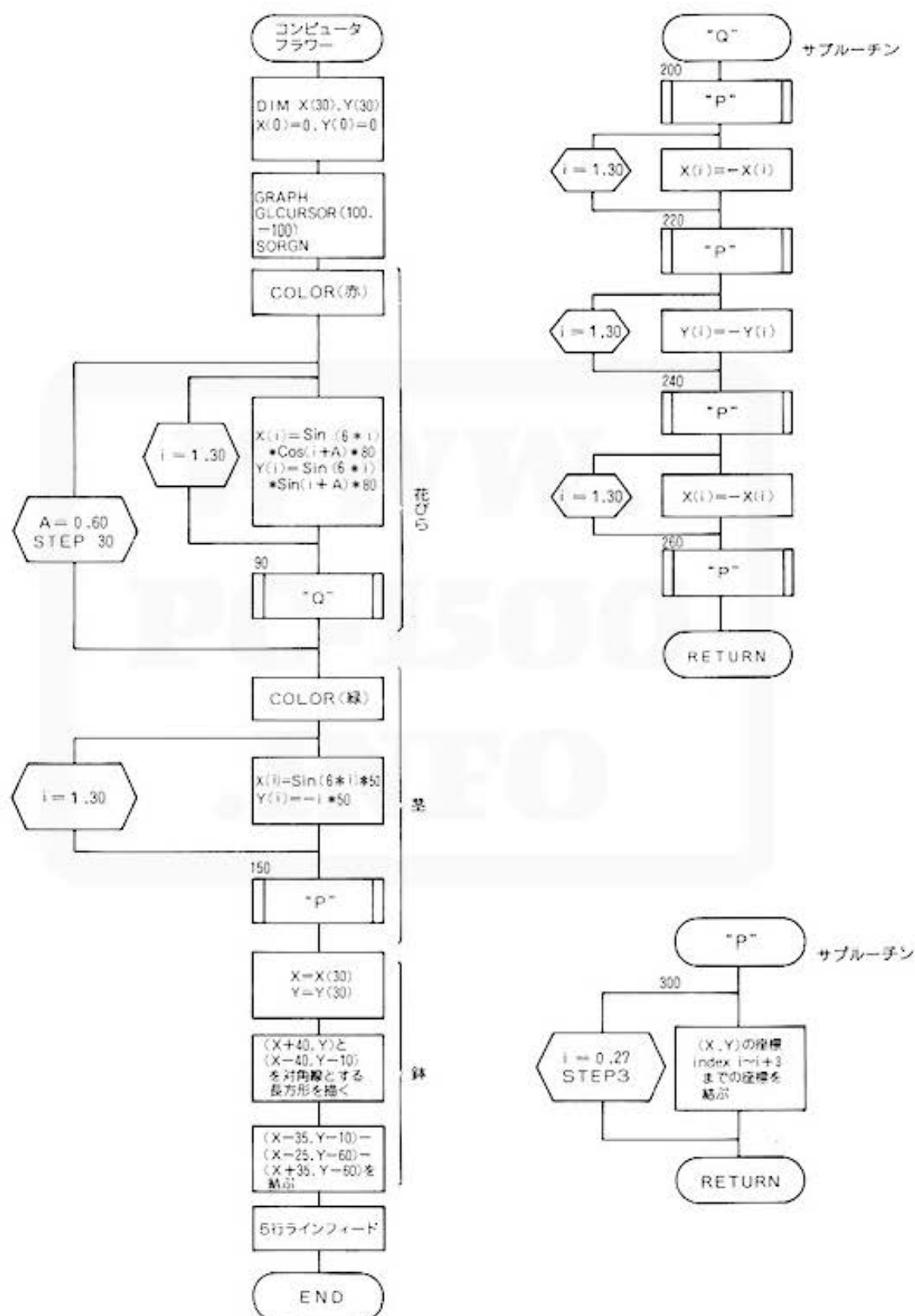
ステップ	キー入力	表示	備考
1	<b>DEF</b> <b>A</b>	>	花の図柄出力

PROGRAM  
TITLE

コンピュータ・フラワー

2

## ■フローチャート



PROGRAM  
TITLE

## コンピュータ・フラワー

3

## ■プログラムリスト ※文字の大きさは18字/行

```

10:"A":CLEAR:DIM
  X(30),Y(30):X(
  0)=0:Y(0)=0
20:GRAPH
30:GLCURSOR(100,
  -100):SORGN
35:COLOR 3
40:FOR A=0 TO 60
  STEP 30
50:FOR I=1 TO 30
60:X(I)=SIN(6*I)
  *COS(I+A)*80
70:Y(I)=SIN(6*I)
  *SIN(I+A)*80
80:NEXT I
90:GOSUB "Q"
100:NEXT A
105:COLOR 2
110:FOR I=1 TO 30
120:X(I)=SIN(6*I)
  *50
130:Y(I)=-I*5
140:NEXT I
150:GOSUB "P"
155:X=X(30):Y=Y(30)
  )
160:LINE(X+40,Y)-
  (X-40,Y-10),0,
  0,B
170:LINE(X-35,Y-1
  0)-(X-25,Y-60)
  -(X+25,Y-60)-(
  X+35,Y-10)
180:TEXT:LF 5:END
200:"Q"GOSUB "P"
210:FOR I=0 TO 30:X
  (I)=-X(I):NEXT
  I
220:GOSUB "P"
230:FOR I=0 TO 30:Y
  (I)=-Y(I):NEXT
  I
240:GOSUB "P"
250:FOR I=0 TO 30:X
  (I)=-X(I):NEXT
  I
260:GOSUB "P"
270:RETURN
300:"P"FOR I=0 TO 2
  7STEP 3
310:LINE(X(I),Y(I)
  )-(X(I+1),Y(I
  +1))-(X(I+2),Y
  (I+2))-(X(I+3)
  ,Y(I+3))
320:NEXT I
330:RETURN

```

メモリー内容	
A	FOR文の変数
B	✓
C	
D	
E	
F	
G	
H	
I	FOR文の変数
J	
K	
L	
M	
N	
O	
P	
Q	
R	
S	
T	
U	
V	
W	
X	X(30)の置き換え
Y	Y(30)の置き換え
Z	
X(30)	X座標
Y(30)	Y座標

**概要**

幾何学模様を描いて、変化の度合いを知りたい時にはこんなプログラムを作ってみるのも一興でしょう。連続した正三角形の列は、アンモナイトの生長過程のようにも見えます。

**■使い方(遊び方)**

角度、増分、個数を様々なパターンで入力することによって描かれる模様の変化をお楽しみください。

**■例題**

(例1) 個数30  
           角度10  
           増分3.5  
           | を入力した場合の作画(プリント出力参照)

(例2) 個数35  
           角度20  
           増分3  
           | を入力した場合の作画

**■内容(計算内容など)**

$$R = R + K$$

(Rは増分の累計で初期値5、入力増分Kの値をワンパターン各に足し込みます。)

$$T = T + S$$

(Tは角度の累計で初期値-S 入力角度Sの値をワンパターン各に足し込みます。)

$$X1 = R * \sin T$$

$$Y1 = R * \cos T$$

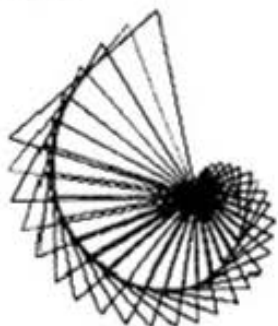
$$X2 = R * \sin (T + 60)$$

$$Y2 = R * \cos (T + 60)$$

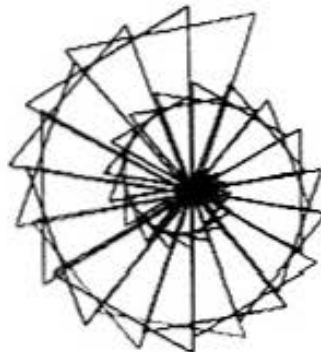
(0, 0) - (X1, Y1) - (X2, Y2) - (0, 0) を直線で結びます。

上記のパターンを入力個数：N回繰り返し作画します。

## ■プリント出力

(実際はカラープリント)  
(出力です。P.121参照) (例1)

(例2)



## ■プログラムリスト ※文字の大きさは18字/行

```

10: "A": GRAPH ;
  RANDOM
20: GLCURSOR (120,
  -200): SORGN
30: INPUT "KOSU=";
  N
40: INPUT "KAKUDO=";
  S
50: INPUT "ZOUBUN=";
  K
60: T=-S: R=5
70: FOR I=1 TO N
75: COLOR (RND 4-1
  )
80: R=R+K: T=T+S
90: X1=R*SIN T: Y1=
  R*COS T
100: X2=R*SIN (T+60
  ): Y2=R*COS (T+
  60)
110: LINE (0,0)-(X1
  ,Y1)-(X2,Y2)-(
  0,0)
120: NEXT I
130: END

```

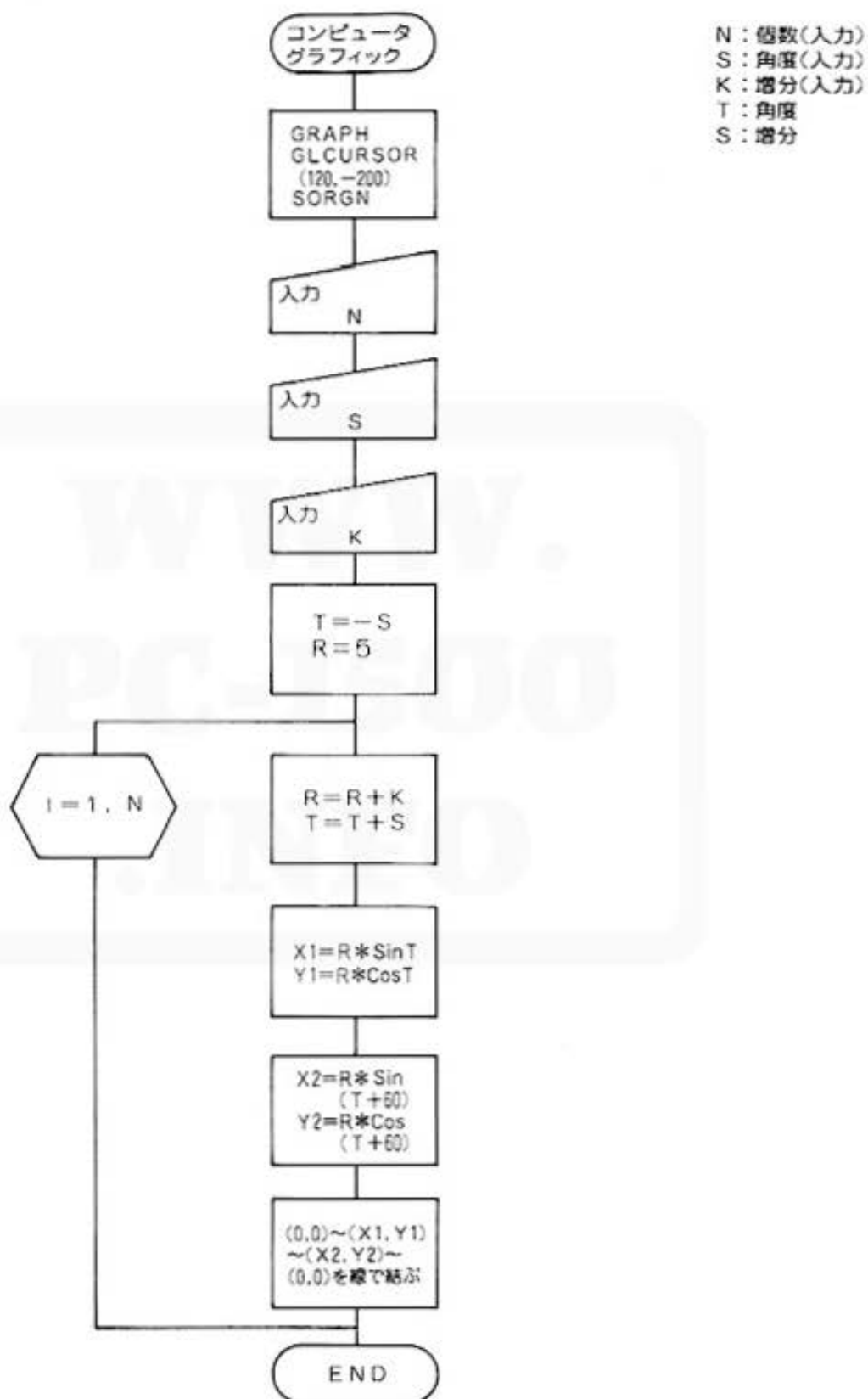


PROGRAM  
TITLE

## コンピュータ・グラフィック

3

## ■フローチャート



PROGRAM  
TITLE

コンピュータ・グラフィック

4

## ■キー操作手順

ステップ	キー入力	表示	備考
1	DEF A	KOSU=	
2	30 ENTER	KAKUDO=	
3	10 ENTER	ZOUBUN=	
4	3.5 ENTER	>	
<hr/>			
1	DEF A	KOSU=	
2	35 ENTER	KAKUDO=	
3	20 ENTER	ZOUBUN=	
4	3 ENTER	>	

メモリー内容			
A		X1	作画X座標の1
B		Y1	＼ Y ＼ の1
C		X2	＼ X ＼ の2
D		Y2	＼ Y ＼ の2
E			
F			
G			
H			
I	√		
J			
K	増分(INPUT)		
L			
M			
N	個数(INPUT)		
O			
P			
Q			
R	増分(計算値)		
S	角度(INPUT)		
T	角度(計算値)		

## エラーコードとエラー内容

エラーコード	エ ラ ー 内 容
1	文法エラー。 プログラムでのみ実行可能な命令をマニュアル操作で、マニュアル操作でのみ実行可能な命令をプログラムで実行しようとした。
2	NEXT命令に対するFOR命令がない。 RETURN命令に対するGOSUB命令がない。
4	READ命令に対するDATAがない。
5	すでに宣言されている配列変数名を再度宣言している。
6	配列変数をDIM命令で宣言せずに使っている。
7	変数名が不適合。(数値変数に文字を代入しようとした場合など)。
8	配列変数の指定が3次元以上になっている。
9	配列変数の添字がDIM命令で宣言した配列の大きさを越えている。
10	変数をメモリー上に確保できない。
11	GOSUB命令、THEN命令などで指定されたラインがプログラム上に存在しない。
12	USING命令におけるフォーマット指定エラー。(指定の形が正しくない)
13	プログラムエリアの容量オーバーあるいはリザーブエリアの容量オーバー。
14	FOR-NEXT命令の段数が深く、スタックエリアの容量をオーバーする。 データや計算命令(ファンクション)のスタック容量オーバー。
15	GOSUB命令の段数が深く、スタックエリアの容量をオーバーする。 文字列の加算や大小比較において、その文字合計が80バイトを越える。
16	指数部が99を超える数値が指定されている。(10E99など) &FFFFを超える16進数値が指定されている。
17	式においてデータの形式が不適合(1+"A"、A\$*2など)
18	演算式においてデータ数が正しくない。(SIN(30, 60)など)
19	指定の数値が規定されている範囲から外れている(DIM A(256)など)
20	固定変数を@( ) あるいは@\$ ( ) で指定するとき、( ) の指定がない。
21	変数でなければならないところが、数式などで指定されている。
22	プログラムを計算機に転送するとき、転送するエリアがない。
23	時刻の設定エラー。
26	指定されているモードで実行できない命令を実行しようとした。
27	指定した命令、関数がない。(オプション機器を本体から切り離した場合に生じることがあります。)
28	" " では含まれた中に命令や関数のコードを挿入している。あるいはINPUT命令やAREAD命令にて文字変数に命令や関数のコードを代入しようとした。
30	プログラム入力時、ラインナンバーが65539を超えている。 (65280~65539はERROR 1)
32	INPUT命令が実行されているとき、GCURSOR命令で152~155ポイントが指定されていて、キー入力した内容を表示する桁がない。

エラーコード	エ ラ ー 内 容
36	U S I N G 命令で指定されたフォーマットにしたがってデータあるいは文字を表示できない。
37	計算結果が9.9999999999 E 99よりも大きい(オーバーフローエラー)
38	除数が0の除算を実行。
39	不合理な演算をおこなった。(L N - 30、A S N 1.5など)
177 } 181	プログラム入力により、すでに定義されていたデータ領域が破壊された。
0 224 } 241	I N P U T 命令あるいはA R E A D 命令実行時に入力データの終了がB A S I C 命令などになっている。
テープレコーダーに関するエラー	
40	式の指定が不適切
42	記録されているプログラムが大きすぎて、計算機のプログラムエリアに転送し切れない。
43	C L O A D ? で照合内容の不一致が発生した。 データの転送において、I N P U T # 命令で指定された変数の形と記録されているデータの形が一致しない。
44	チェックサムエラーが発生した。
プリンタに関するエラー	
70	ペンが $-2048 \leq X, Y \leq 2047$ の座標範囲を越える。
71	T E X T モードで10.24cm以上紙を逆に進めようとした。
72	T A B 命令で指定した値が不適当。
73	G R A P H モードで使用できない命令、あるいはT E X T モードで使用できない命令を実行しようとした。
74	L I N E あるいはR L I N E 命令で( , ) で指定する座標の数が多い。
76	T E X T モードにおいて、L P R I N T 命令で計算結果(数値)が1行で印字できない。
78	次の理由により、L P R I N T、L I N E などの命令が実行できない。 ① ペン交換中 ② ローバッテリーチェック(E R R O R 80)により掛ったプリンタロックが解除されていない。
79	カラー信号が出ていない。
80	ローバッテリーチェック(プリンタの内部電池の電圧が低下し、プリンタが正しく働かない恐れがあるときに、プリンタをロックします。C E -150 の取扱説明書を参照してください。)

エラー1が出た場合以下のことに留意してください。

(1) 入力に関して

◎ 入力ミスを訂正するような場合

```
10 PRINT A$ ENTER
↓
10 GPRINT A$ ENTER
    ↑
    AをGに変更した
```

◎ CURSOR命令をGCURSOR命令に変更するような場合

```
10 CURSOR 10 ENTER
↓
10 GCURSOR 10 ENTER
    ↑
    Gを挿入した
```

◎ GPRINT 入力時 **G** **DEF** **W** と操作したような場合

以上のような方法によって修正あるいは入力された場合、計算機には1つの命令として判断されません。このような場合は、その命令のスペルをアルファベットキーにより最初から再入力していただく必要があります。

(例)

	10: PRINT A\$
	↑ カーソル点滅
▶	10 PRINT A\$
SHIFT DEL ◀	10 PRINT A\$
SHIFT DEL ◀	10 PRINT A\$
SHIFT INS ▶ ..... SHIFT INS ▶	10 A\$
G P R I N T ENTER	10 GPRINT A\$

なお、命令が正しく入力されているかどうかはカーソルキーでチェックすることができます。

(正しい入力の場合)

```
↑ カーソル点滅
GPRINT A$
↓
▶ GPRINT A$
```

(間違った入力の場合)

```
GPRINT A$
↓
▶ GPRINT A$
↓
▶ GPRINT A$
```

# キャラクタ・コード表

16  
8  
111  
148  
128

上位ビット(上位桁) →

		0000	0001	0010	0011	0100	0101	0110	0111
	16進	0	1	2 <sub>32</sub>	3 <sub>48</sub>	4	5	6	7
0000	0				0	@	P		p
0001	1			3 <sub>3</sub> !	1	A	Q	a	q
0010	2			"	2	B	R	b	r
0011	3			#	3	C	S	c	s
0100	4			\$	4	D	T	d	t
0101	5			%	5	E	U	e	u
0110	6			&	6	F	V	f	v
0111	7			[	7	G	W	g	w
1000	8			(	8	H	X	h	x
1001	9			)	9	I	Y	i	y
1010	A			*	:	J	Z	j	z
1011	B			+	;	K	✓	k	ı
1100	C			,	<	L	¥	l	ı
1101	D			-	=	M	π	m	ı
1110	E			.	>	N	^	n	~
1111	F			/	?	O	_	o	■

↓ (低位桁) 7ビット

## 関数・命令(省略形)一覧

## 1. 基本関数

関 数	省 略 形	備 考
SIN	S I.	$\sin$
COS		$\cos$
TAN	T A.	$\tan$
ASN	A S.	$\sin^{-1}$
ACS	A C.	$\cos^{-1}$
ATN	A T.	$\tan^{-1}$
LOG	L O.	$\log_{10} X$
LN		$\log_e X$
EXP	E X.	$e^x$
^		べき乗
√ (SQR)	S Q.	開平
DEG		度分秒 → 10進数度変換
DMS	D M.	10進数度 → 度分秒変換
INT		整数化 (インテジャー) INT (10/3) <span style="border: 1px solid black; padding: 0 2px;">ENTER</span> → 3
ABS	A B.	絶対値 (アブソリュート) ABS (-5 * 2) <span style="border: 1px solid black; padding: 0 2px;">ENTER</span> → 10
SGN	S G.	符号関数 (シグナム)
π (PI)		円周率を指定 (π = 3.141592654)



## 2. BASIC 命令

(※……プログラムで実行可能な命令)  
(★……マニュアル操作で実行可能な命令)

命 令	省 略 形	一 般 形	備 考
※ LET (レット)	LE.	1) LET 数値変数-式 2) LET 文字変数-"文字" 3) LET 文字変数-文字変数	LETは IF 文に続く場合以外は省略可
※ INPUT (インプット)	I. } INPU.	1) INPUT 変数, 変数, ... 2) INPUT "文字", 変数, "文字", 変数, ... 3) INPUT "文字": 変数, "文字": 変数, ...	入力命令 データを入力
※ PRINT ★ (プリント)	P. } PRIN.	<div style="display: flex; flex-direction: column; align-items: center;"> <div style="display: flex; align-items: center;"> <div style="text-align: right;">1) PRINT</div> <div style="border-left: 1px solid black; padding-left: 5px; text-align: center;">             式 "文字" 文字変数           </div> </div> <div style="display: flex; align-items: center;"> <div style="text-align: right;">2) PRINT</div> <div style="display: flex; align-items: center;"> <div style="border-left: 1px solid black; padding-left: 5px; text-align: center;">             式 "文字" 文字変数           </div> <div style="margin: 0 5px;">,</div> <div style="border-left: 1px solid black; padding-left: 5px; text-align: center;">             式 "文字" 文字変数           </div> </div> </div> <div style="display: flex; align-items: center;"> <div style="text-align: right;">3) PRINT</div> <div style="display: flex; align-items: center;"> <div style="border-left: 1px solid black; padding-left: 5px; text-align: center;">             式 "文字" 文字変数           </div> <div style="margin: 0 5px;">;</div> <div style="border-left: 1px solid black; padding-left: 5px; text-align: center;">             "文字" 変数           </div> <div style="margin: 0 5px;">;</div> <div style="border-left: 1px solid black; padding-left: 5px; text-align: center;">             .....              ...;              "文字" 変数           </div> </div> </div> <div style="margin-top: 10px;">4) 上記一般形(1), 3)の最後に ; をつけた形</div> </div>	出力命令 指定された内容を表示します。
※ END (エンド)	E. EN.	END	プログラム終了命令
※ DEGREE ★ (ディグリー)	DE. } DEGRE.	DEGREE	角度単位指定 "度"単位指定〔°〕
※ RADIAN ★ (ラディアン)	RAD. } RADIA.	RADIAN	角度単位指定 "ラディアン"単位指定〔rad〕
※ GRAD ★ (グラード)	GR. GRA.	GRAD	角度単位指定 "グラード"単位指定〔g〕
※ GOTO ★ (ゴートゥ)	G. } GOT.	<div style="display: flex; flex-direction: column; align-items: center;"> <div style="display: flex; align-items: center;"> <div style="text-align: right;">(1) GOTO</div> <div style="border-left: 1px solid black; padding-left: 5px; text-align: center;">             式 "文字" 文字変数           </div> </div> <div style="margin-top: 10px;">(2) GOTO (単独の場合はマニュアル操作でのみ実行可能)</div> </div>	ジャンプ命令 (プログラム実行開始命令) ・指定されたラインあるいはラベルへ実行を移す。 ・指定されたラインからプログラムを実行する。

命 令	省 略 形	一 般 形	備 考
※ IF (イフ)		IF { 条件式 } 実行文 式	判断命令 与えられた条件を判断し、実行を次の実行文か、あるいは次のラインへ移します。
※ THEN (ゼン)	T. } THE.	IF 文における実行文として定義します。 THEN { 式 } "文字" 文字変数	ジャンプ命令 IF 文における実行文として定義することのみ可能。
※ FOR (フォー) TO (トゥー) STEP (ステップ)	F. FO.  STE.	(1) FOR 数値変数-式, TO 式, (2) FOR 数値変数-式, TO 式, STEP 式, 式 <sub>1</sub> : 初期値, 式 <sub>2</sub> : 最終値 式 <sub>3</sub> : 増(減)分	FOR-NEXTループの始まり NEXT命令と対で使用
※ NEXT (ネクスト)	N. } NEX.	NEXT 数値変数 数値変数はFOR命令に続く数値変数と一致していなければならない。	FOR-NEXTループの末端を示します。 FOR文と対で使用。
※ GOSUB (ゴースブ)	GOS. GOSU.	GOSUB { 式 } "文字" 文字変数	サブルーチンジャンプ命令 指定されたラインあるいはラベルへ実行を移し、そこからのプログラムをサブルーチンとして実行。 RETURN命令と対で使用。
※ RETURN (リターン)	RE. } RETUR.	RETURN	復帰命令 サブルーチンの復帰命令で、GOSUBの次のステートメントへ実行が復帰します。
※ DIM ★ (ディメンジョン)	D. DI.	(1) DIM 変数名(式) (2) DIM 変数名(式)*式, (3) DIM 変数名(式 <sub>1</sub> , 式 <sub>2</sub> ) (4) DIM 変数名(式 <sub>1</sub> , 式 <sub>2</sub> )*式, 変数名: A, B, CS, D\$ など ( ) : 配列の大きさ、次元指定 式 <sub>1</sub> : 桁数指定 (文字変数のみ)	変数の定義 配列変数の変数名、大きさなどを定義し、プログラム・データエリアに確保する。

命 令	省 略 形	一 般 形	備 考
※ ★ USING (ユージング)	U. } USIN.	(1) USING “#…#, #…#^” (2) USING “&&&……&&&” (3) PRINT USING “フォーマット”;…  (4) USING  (5) PRINT USING ;…	フォーマット指定命令 数値データ、文字データの出力フォーマット指定  フォーマット指定解除
※ ★ CURSOR (カーソル)	CU. } CURSO.	(1) CURSOR 式 (式の値は 0~25)  (2) CURSOR	表示開始位置指定 式の値で表示ポジションを指定 指定を解除。
※ ★ WAIT (ウェイト)	W. } WAI.	(1) WAIT 式 (式の値は 0~65535)  (2) WAIT	表示時間指定 PRINT 命令によるプログラム停止時間を指定。 指定を解除。 (停止時間無限)
※ ★ PAUSE (ポーズ)	PA. } PAUS.	一般形は PRINT 命令と同じ形をとります。	出力命令 指定された内容を約 0.85秒表示したのちプログラムを実行。
※ ★ CLS (クリアーズ)		CLS	表示クリア命令 表示を消去します。
※ ★ GCURSOR (グラフィック カーソル)	GC. } GCURSO.	GCURSOR 式 (式の値は 0~155 あるいは & 0~ & 9 B)	表示開始位置指定 表示ポジションをドットの単位で指定。
※ ★ GPRINT (グラフィック プリント)	GP. } GPRIN.	(1) GPRINT “○○ ○○ ○○ …” (“ ” の中は16進数字) (2) GPRINT ○;○;… (3) GPRINT &○;&○;…	出力命令 ドットパターンで指定された内容を表示します。
※ ★ READ (リード)	REA.	READ 変数, 変数, …	データの読み込み命令 DATA 命令に続いて指定されているデータを指定の変数に入れる。

命 令	省 略 形	一 般 形	備 考
※ DATA (データ)	DA. DAT.	DATA 式, 式, ...	データ指定 READ 命令に続く 変数に与えるデータ を指定。
※ RESTORE (リストア)	RES. } RESTOR.	1) RESTORE 式 2) RESTORE	READ 命令により 読み込まれるデータ の順番を変える命令
※ ON GOTO (オン ゴートゥー)	ONG. O. G. } O. GOT.	ON 式 GOTO 式, 式, 式, ...	ジャンプ命令 ON に続く式の値に よりジャンプ先を指 定する。
※ ON GOSUB (オン ゴースブ)	ONGOS. O. GOS. O. GOSU.	ON 式 GOSUB 式, 式, 式, ...	サブルーチンジャン プ命令 ON に続く式の値に よりジャンプ先を指 定。
※ ON ERROR (オン エラー)	ONER. O. ER. } O. ERRO.	ON ERROR (ON ERROR GOTO 式)	エラー処理のための 命令
※ REM (リマーク)		REM 注釈	プログラムの中に非 実行文 (注釈) を指 定するための命令。
※ ARUN (オートラン)	ARU.	ARUN	電源オン時、自動的 にプログラムの実行 を開始する命令。
※ STOP (ストップ)	S. } STO.	STOP	プログラムの実行を 停止する命令。
※ AREAD (オートリード)	A. } AREA.	AREAD 変数	定義付けキーによる プログラム実行開始 時に、表示内容を指 定された変数に入れ る。
※ CLEAR ★ (クリア)	CL. } CLEA.	CLEAR	データ (変数) をす べて消去する命令。
※ BEEP ★ (ビーブ)	B. } BEE.	BEEP 式, 式, 式,  BEEP ON BEEP OFF	音命令 音の高さや、長さを 指定。 音発生機能をオン 音発生機能をオフ

命 令	省 略 形	一 般 形	備 考
※ ★ TR ON (トレース オン)	TR. TRO.	TR ON	デバッグをおこなう モードを指定。
※ ★ TR OFF (トレース オフ)	TROF.	TR OFF	デバッグをおこなう モードを解除。
※ ★ LOCK (ロック)	LOC.	LOCK	[MODE] キーで切替える モードをロックする。
※ ★ UNLOCK (アンロック)	UN. UNLOC.	UNLOCK	モードのロックを解 除する。
※ ★ RANDOM (ランダ マイズ)	RA. RANDO.	RANDOM	RND 命令に先だっ て乱数のタネを植え つけるものです。

### 3. コマンド(マニュアル操作でのみ実行可能)

命 令	省 略 形	一 般 形	備 考
★ NEW (ニュー)		(1) NEW (2) NEW 0 (PROモードでのみ実行可能)	PROモードでプロ グラムおよび全変数 をクリア。 RESERVE モー ドでリザーブ内容を クリア。
★ RUN (ラン)	R. RU.	(1) RUN (2) RUN 式 (3) RUN { “文字” 文字変数 }	プログラム実行開始 命令。 RUN モードで有効
★ LIST (リスト)	L. LIS.	一般形はRUN 命令と同様に定義	プログラムのリスト をおこなう命令。 PROモードで有効
★ CONT (コンティ ニュー)	C. CON.	CONT	一時停止しているプ ログラムの実行を再 開する命令。 RUN モードで有効

## 4. BASIC関数

	命 令	省 略 形	一 般 形	備 考
※ ★	RND (ランダム)	RN.	RND 式	乱数を得る命令。
※ ★	AND (アンド)	AN.	式 AND 式	論理積
※ ★	OR (オア)		式 OR 式	論理和
※ ★	NOT (ノット)	NO.	NOT 式	否 定
※ ★	LEFT\$ (レフトドル)	LEF. LEFT.	LEFT\$ (文字変数, 数式)	指定された文字列の 左側から、指定された 桁数の文字を取り出す。
※ ★	RIGHT\$ (ライトドル)	RI. }	RIGHT\$ (文字変数, 数式)	指定された文字列の 右側から、指定され た桁数の文字を取り 出す。
※ ★	MID\$ (ミッドドル)	MI. MID.	MID\$ (文字変数, 数式, , 数式, )	指定された文字列の中 間の文字を取り出す。
※ ★	LEN (レングス)		LEN   "文字"     文字変数	指定された文字列の 文字数を求めます。
※ ★	STR\$ (ストリング ドル)	STR.	STR\$ 数式 (A\$-STR\$ A)	数値を文字列に変換
※ ★	VAL (バリュー)	V. VA.	VAL   "文字"     文字変数	文字列を数値に変換
※ ★	ASC (アスキー)		ASC   "文字"     文字変数	文字などのキャラク タをアスキーコード に変換。
※ ★	CHR\$ (キャラクタ ドル)	CH. CHR.	CHR\$ 数式	アスキーコードを文 字などのキャラクタ に変換。
※ ★	STATUS (ステータス)	STA. }	STATUS 0 STATUS 1	プログラム・データエ リアのプログラム以外 のバイト数、プログラ ムのバイト数などの 呼び出し。
※ ★	MEM (メモ)	M. ME.	MEM	STATUS 0と同じ プログラム・データ エリアのプログラム 以外のバイト数を 表示。

命 令	省 略 形	一 般 形	備 考
※ ★ POINT (ポイント)	POI. POIN.	POINT 式 (式の値は 0~155 あるいは & 0~ & 9B) 例 A-POINT 56	ドットパターン読み 出し命令 指定されたポイント に表示されているド ットパターン情報を 読み出す。

## 5. その他

命 令	省 略 形	一 般 形	備 考
※ ★ TIME (タイム)	TI. TIM.	(1) TIME-月日時、分秒  2) TIME	タイム関数 月日時、分秒をセッ ト。 日付、時刻呼び出し
※ ★ INKEY\$ (イン キードル)	INK. } INKEY.	文字変数-INKEY\$	INKEY\$ 命令実 行時にキーが押され ていればその内容を 読み込む。

## 6. プリンタ関係命令

命 令	省 略 形	一 般 形	備 考
※ ★ TEXT (テキスト)	TEX.	TEXT	TEXT モード指定 文字や数字を印字す るモード。
※ ★ GRAPH (グラフィック)	GRAP.	GRAPH	GRAPH モード指 定 グラフなど図形を書 くモード。
※ ★ LPRINT (ライン プリント)	LP. } LPRIN.	一般形は PRINT 命令と同じ形をとります。	出力命令 指定された内容を印 字します。
※ ★ CSIZE (キャラクタ サイズ)	CSI. CSIZ.	CSIZE 式 (式の値は 1~9)	印字する文字の大き さを指定。
※ ★ ROTATE (ローテイト)	RO. } ROTAT.	GRAPH モードでのみ有効。 ROTATE 式 (式の値は 0~3)	印字する文字の向き (印字方向)を指定。
※ ★ COLOR (カラー)	COL. COLO.	COLOR 式 (式の値は 0~3)	文字などの色を指定



	命 令	省 略 形	一 般 形	備 考
※ ★	TEST (テスト)	TE. TES.	TEST	色のチェック TEST 命令実行で 5mm角の正方形をそ れぞれの色で書く。
※ ★	TAB (タブ)		TEXT モードでのみ有効。 (1) TAB 式 (2) LPRINT TAB 式; ...	ペンの位置を指定
※ ★	LF (ライン フィード)		TEXT モードでのみ有効。 LF 式	紙送り命令 式で示す行数分の紙 送りをおこないます。
※ ★	GLCURSOR (グラフィック ライン カーソル)	GL. } GLCURSO.	GRAPH モードでのみ有効。 GLCURSOR (式 <sub>1</sub> , 式 <sub>2</sub> )	ペン位置を原点から X 軸、Y 軸方向に移 動させる命令。
※ ★	SORGN (セット オリジン)	SO. } SORG.	GRAPH モードでのみ有効。 SORGN	現在あるペンの位置 を新しい原点に指定 する命令。
※ ★	LINE (ライン)	LIN.	GRAPH モードでのみ有効。 (1) LINE (式 <sub>1</sub> , 式 <sub>2</sub> )-(式 <sub>3</sub> , 式 <sub>4</sub> ) (2) LINE (式 <sub>1</sub> , 式 <sub>2</sub> )-(式 <sub>3</sub> , 式 <sub>4</sub> ), 式 <sub>5</sub> , 式 <sub>6</sub> (3) LINE (式 <sub>1</sub> , 式 <sub>2</sub> )-(式 <sub>3</sub> , 式 <sub>4</sub> ), 式 <sub>5</sub> , 式 <sub>6</sub> , B 式 <sub>5</sub> : 線の種類を指定 式 <sub>6</sub> : 色を指定 B: 四角形を指定 (4) LINE (式 <sub>1</sub> , 式 <sub>2</sub> )-(式 <sub>3</sub> , 式 <sub>4</sub> )-... ... (式 <sub>11</sub> , 式 <sub>12</sub> )	線を書く命令です。
※ ★	RLINE (レラティブ ライン)	RL. } RLIN.	GRAPH モードでのみ有効。 一般形は LINE 命令と同じ形となります。	ペン位置を原点とし て、線を書く命令で す。
※ ★	LLIST (ライン リスト)	LL. } LLIS.	(1) LLIST <input type="button" value="ENTER"/> (2) LLIST 式 <input type="button" value="ENTER"/> (3) LLIST 式 <sub>1</sub> , 式 <sub>2</sub> <input type="button" value="ENTER"/> (4) 式 <sub>1</sub> あるいは 式 <sub>2</sub> を省略した形	リスト命令 プリンタによりプロ グラムのリストアウ トをおこなう命令。

## 7. テープレコーダーに関する命令

	命 令	省 略 形	一 般 形	備 考
※ ★	CSAVE (カセット セーブ)	CS. } CSAV.	(1) CSAVE (CSAVE-1) (2) CSAVE "ファイル名" (CSAVE-1 "ファイル名")	記録命令 プログラムやリザー ブ内容をテープに記 録する命令。

命 令	省 略 形	一 般 形	備 考
★ CLOAD (カセット ロード)	CLO. CLOA.	(1) CLOAD (CLOAD-1) (2) CLOAD "ファイル名" (CLOAD-1 "ファイル名")	転送命令 プログラムあるいは リザーブ内容をテー プから計算機に転送 する命令。
★ MERGE (マージ)	MER. MERG.	一般形はCLOAD命令と同じ形となります。 [この命令では、それまで記憶していたプ ログラムはそのまま保持し、その後、新 しく読み込んだプログラムが記憶される。]	転送命令 プログラムをテーブ から計算機に転送す る命令。
※ CHAIN (チェーン)	CHA. CHAI.	(1) CHAIN "ファイル名" (CHAIN-1 "ファイル名") (2) CHAIN "ファイル名", 式 (CHAIN-1 "ファイル名", 式) (3) 上記(1), (2)の"ファイル名"を省略した形 (2の形でファイル名を省略したとき、式がラベル 指定の場合は1と同じ形式となります。)	転送命令 プログラムの中で使 用する命令。プログ ラムをテープから読 み込んで(転送して) そのプログラムを実 行。
★ CLOAD? (カセット ロード?)	CLO. ? CLOA. ?	(1) CLOAD ? (CLOAD ?-1) (2) CLOAD ? "ファイル名" (CLOAD ?-1 "ファイル名")	照合命令 計算機内のプログラ ムあるいはリザーブ 内容とテープに記録 されている内容との 照合をおこなう命令。
※ PRINT # ★ (プリント クロスハッチ)	P. # } PRIN. #	(1) PRINT # 変数名, 変数名, ... (PRINT #-1, 変数名, 変数名, ...) (2) PRINT # "ファイル名"; 変数名, 変数名, ... (PRINT #-1, "ファイル名"; 変数名, 変数名, ...)	データ記録命令 計算機に記憶されて いるデータをテーブ に記録する命令。
※ INPUT # ★ (インプット クロスハッチ)	I. # } INPU. #	一般形はPRINT # 命令と同じ形になりま す。	データ転送命令 テーブに記録されて いるデータを指定さ れた変数に転送する 命令。
※ RMT OFF ★ (リモート オフ)	RM. OF.	RMT OFF	REM 1 端子のリモ ート機能を解除する 命令。
※ RMT ON ★ (リモート オン)	RM. O.	RMT ON	REM 1 端子のリモ ート機能をふたたび セットする命令。

## 索引

(ページ)	(ページ)
アスキーコード.....73	変数.....18、42
一次元配列.....43	メインルーチン.....37
エラー.....15、26	メモリー.....18、44
エラーコード.....167	モード.....10
エラー処理.....26、57	文字変数.....18、42
オートパワーオフ.....9	文字列.....74
オールリセットスイッチ.....4	ライン.....21
音発生機能.....62	ラインナンバー.....22
カーソル.....14	ラベル.....30
角度指定.....16	乱数.....60
記録.....96	ランモード.....10
クリア.....46	リザーブエリア.....45
原点.....88	リザーブ機能.....76
サブルーチン.....37	リザーブモード.....10
ジャンプ.....29	リスト.....25、92
条件式.....30	論理積.....64
照合.....101	論理和.....64
シンボル.....115	ABS.....16
数値変数.....18、42	ACS.....16
スタックエリア.....41	AND.....64、66
ステートメント.....22	AREAD.....59
定義付け.....39	ARUN.....58
定義付けキー.....40	ASC.....73、74
データ専用エリア.....45	ASN.....16
デバッグ.....68	ATN.....16
転送.....97	BEEP.....62
二次元配列.....43	CHAIN.....99
バイト.....5、41、46	CHR\$.....74
配列変数.....42	CLEAR.....46
判断.....30	CLOAD.....97
否定.....67	CLOAD?.....101
表示ポジション.....50	CLS.....51
フォーマット指定.....48	COLOR.....86
プレイバック.....14	CONT.....63
フローチャート.....33	COS.....16
プログラミング.....20	CPU.....5
プログラムモード.....10	CSAVE.....96
プログラム・データエリア.....45	CSIZE.....85
プロンプト記号.....9	CURSOR.....50
	DATA.....55
	DEG.....16

	(ページ)		(ページ)
DEGREE.....	16	PRINT#.....	102
DIM.....	42	RADIAN.....	16
DMS.....	16	RAM.....	5
END.....	22	RANDOM.....	60
EXP.....	16	READ.....	55
FOR.....	35	REM.....	57
GCURSOR.....	51	RESTORE.....	55
GLCURSOR.....	88	RETURN.....	37
GOSUB.....	37	RIGHT\$.....	70
GOTO.....	29	RLINE.....	91
GPRINT.....	52	RMT OFF.....	104
GRAD.....	16	RMT ON.....	104
GRAPH.....	84	RND.....	60
IF.....	30	ROM.....	5
INKEY\$.....	61	ROTATE.....	85
INPUT.....	22, 26	RUN.....	25
INPUT#.....	102	SGN.....	16
INT.....	16	SIN.....	16
LEFT\$.....	70	SORGN.....	88
LEN.....	72	SQR.....	16
LET.....	22	STATUS.....	63
LF.....	87	STEP.....	35
LINE.....	89	STOP.....	58
LIST.....	25	STR\$.....	73
LLIST.....	92	TAB.....	86
LN.....	16	TAN.....	16
LOCK.....	67	TEST.....	86
LOG.....	16	TEXT.....	84
LPRINT.....	84	THEN.....	31
MEM.....	63	TIME.....	61
MERGE.....	98	TO.....	35
MIDS.....	71	UNLOCK.....	67
NEW.....	4, 23	USING.....	48
NEXT.....	35	VAL.....	73
NOT.....	65, 67	WAIT.....	50
ON ERROR.....	57		
ON GOSUB.....	57		
ON GOTO.....	56		
OR.....	64, 66		
PAUSE.....	50, 51		
PI.....	16		
POINT.....	54		
PRINT.....	22, 27, 47		

■使い方：

PROGRAM TITLE	氏 名
<p>■参 考： (計算式など)</p>	
<p>■例 題：</p>	

PROGRAM  
TITLE

氏 名

■プリント出力(結果):





### ■ 丰一操作手顺：

PROGRAM T I T L E		氏 名	
■プログラムリスト：			
ラインナンバー	ステートメント	ラインナンバー	ステートメント

PROGRAM TITLE			氏 名		
■メモリー内容：					
A		A\$			
B		B\$			
C		C\$			
D		D\$			
E		E\$			
F		F\$			
G		G\$			
H		H\$			
I		I\$			
J		J\$			
K		K\$			
L		L\$			
M		M\$			
N		N\$			
O		O\$			
P		P\$			
Q		Q\$			
R		R\$			
S		S\$			
T		T\$			
U		U\$			
V		V\$			
W		W\$			
X		X\$			
Y		Y\$			
Z		Z\$			

PROGRAM  
TITLE

氏 名

■フローチャート(流れ図):



\* \* \* ✕ ㄟ \* \* \*



巻末の保証書は、本書記載内容で無料修理をさせて頂くことをお約束するものです。

保証期間中に故障が発生した場合は、製品と本書をご持参、ご提示の上、お買上げの販売店にご依頼ください。

お買上げ年月日、販売店名など記入もれがありますと無効となります。必ずご確認いただき、記入のない場合はお買上げの販売店にお申し出ください。

ご転居・ご贈答品などでお買上げ販売店に修理をご依頼できない場合は、製品に同梱しております「シャープサービス・お客様ご相談窓口所在地一覧表」をご覧の上、もよりのサービス会社へご持参ください。保証書は再発行いたしません。たいせつに保管してください。

#### 〈無料修理規定〉

1. 取扱説明書・本体注意ラベルなどの注意書にしたがった正常な使用状態で、保証期間内に故障した場合には、お買上げ販売店、または当社サービス会社が無料修理いたします。
2. 保証期間内でも、次の場合には有料修理となります。
  - (イ) 保証書のご提示がない場合。
  - (ロ) 保証書にお買上げ年月日・お客様名・販売店名の記入がない場合、または字句を書き換えられた場合。
  - (ハ) 使用上の誤り、または不当な修理や改造による故障・損傷。
  - (ニ) お買上げ後に落とされた場合などによる故障・損傷。
  - (ホ) 火災・公害および地震・風水害その他天災地変など、外部に要因がある故障・損傷。
  - (ヘ) 電池の液もれによる故障・損傷。
  - (ト) 消耗品（乾電池）が損耗し取り替えを要する場合。
3. 保証書は日本国内においてのみ有効です。

(THIS WARRANTY CARD IS ONLY VALID FOR SERVICE IN JAPAN.)

★この保証書は本書に明示した期間・条件のもとにおいて無料修理をお約束するものです。したがってこの保証書によってお客様の法律上の権利を制限するものではありませんので、保証期間経過後の修理などにつきましておわかりにならない場合はお買上げの販売店、またはシャープお客様ご相談窓口にお問い合わせください。

修理メモ

# シャープ計算機保証書 (WARRANTY CARD)

品 名 計 算 機

形 名 PC-1500

保証期間 お買上げ日より本体1年間  
(VALIDITY) (FULL 1 YEAR AFTER PURCHASE)

お買上げ日 昭和 年 月 日  
(PURCHASE)

シャープ株式会社 産業機器事業本部  
〒639 11 奈良県大和郡山市美濃庄町492  
電話 (07435)3-5521番



お 客 様	ご芳名	様
	ご住所	〒
	電話番号 ( ) -	
取扱販売店名・住所・電話番号		
印		





# シャープ株式会社

本社 〒545 大阪市阿倍野区長池町22-22

電話 06-621-1221(大代表)

販賣促進課 〒633-11 奈良県大和郡山田町妻道王司452

電話 07435-3-5521(大代表)

# シャープビズネス株式会社

〒545 大阪市阿倍野区長池町22-22 電話 06-621-1221(大代表)

販推部 〒162 東京都新宿区市台八幡町8番地 電話 03-260-1161(大代表)

札幌支店 〒063 札幌市西区二十四軒1条7丁目3番17号 電話 011-641-3631

仙台支店 〒983 宮城県仙台市青野町2丁目8番地の9 電話 0222-94-3247

東京支店 〒130 東京都墨田区石神2丁目12番3号 電話 03-525-5111

千葉支店 〒236 千葉県市川市1丁目5番20号 電話 0472-63-4043

横浜支店 〒225 横浜市中区中町1丁目2番23号 電話 045-751-3215

埼玉支店 〒334 埼玉県大宮市宮前町2丁目107番地2号 電話 0486-63-8169

新潟支店 〒950 新潟市上州町34番番地 電話 0252-83-1795

名古屋支店 〒454 名古屋市中川区山王三丁目5番5号 電話 052-332-2631(大代表)

金沢支店 〒921 石川県石川郡野々市町池崎1056番1号 電話 0762-49-1240

大阪支店 〒556 大阪市東淀川区東淀川1丁目2番9号 電話 06-643-3021(大代表)

京都支店 〒601 京都市南区上鳥羽町48番地 電話 075-681-4361(大代表)

神戸支店 〒658 神戸市東灘区魚崎1丁目11番地16号 電話 078-431-4361(大代表)

広島支店 〒734 広島市安佐南区緑町大字西郷249番地の1 電話 08287-4-4325

高松支店 〒760 香川県高松市木太町1861番地の3 電話 0878-33-4255

福岡支店 〒816 福岡市博多区井明田2丁目12番地の1 電話 092-572-2611

沖縄支店 〒900 沖縄県那覇市腰2丁目10番地の1 電話 0988-61-7360(大代表)

旭川 電話 0116-25-2565) 函 館 電話 0138-55-0232) 釧 路 電話 0154-22-9777)

旭 山 電話 0249-45-4380) 青 森 電話 0177-34-4732) 盛 岡 電話 0196-38-8858)

秋 田 電話 0186-62-0027) 山 形 電話 0226-33-3215) 雄 勝 電話 03-620-9591)

城 西 電話 03-384-7638) 城 南 電話 03-737-1561) 三 菱 電話 0425-84-1220)

松 戸 電話 0473-55-2131) 水 戸 電話 0292-41-0552) 平 塚 電話 0463-54-2624)

静 岡 電話 0542-82-8825) 沼 津 電話 0559-21-3741) 甲 府 電話 0552-26-1035)

宇都宮 電話 0286-37-3575) 前 橋 電話 0272-23-3511) 長 岡 電話 0258-36-7285)

長野 電話 0262-29-4618) 松 本 電話 0263-25-0547) 豊 橋 電話 0532-53-0646)

津 (電話 0592-32-4123) 四日市 電話 0569-51-6703) 岐阜 電話 0582-73-2325)

浜 松 電話 0534-63-1551) 富 山 電話 0764-51-1317) 福 井 電話 0776-27-1900)

阪 神 電話 06-421-2204) 東大阪 電話 0729-94-1519) 南大阪 電話 0722-41-9265)

奈良 電話 07435-3-1423) 和歌山 電話 0734-44-5130) 大 津 電話 0775-24-4436)

总 機 電話 0749-23-8994) 賢 穂 電話 0773-75-3653) 姫 路 電話 0792-66-8411)

東 広 島 電話 0854-28-3081) 松 江 電話 0852-26-5211) 堀 山 電話 0843-51-3435)

岡 山 電話 0862-43-6644) 備 前 電話 0857-22-8278) 徳 山 電話 0834-31-5520)

小 倉 電話 0937-2-3834) 下 関 電話 0832-52-5508) 松 山 電話 0899-71-1145)

徳 島 電話 0886-25-2103) 高 知 電話 0868-83-7078) 長 崎 電話 0956-46-9513)

北九州 電話 093-592-6621) 大 分 電話 0975-36-0289) 熊本 電話 0963-62-9036)

鹿児島 電話 0992-62-1659) 宮 崎 電話 0985-25-2251)