SEGA® パソコン SC-3000 シリーズ BASIC入門

セガ・エンタープライゼス編

初めてだってすぐわかる

SEGA® パソコン SC-3000 シリーズ BASIC入門

セガ・エンタープライゼス編
ダイヤモンド社

おもしろソフト
プログラム集
●ブロックくずしゲーム
●UFO撃墜ゲーム●SEA BATTLE
●ナンバープール●お絵描き教室●幾何学模様
●♪SWEET MEMORIES●マイ算数塾
●2次方程式グラフ●体重診断●バイオリズム
●住所録●株式チャート●ローンの金利計算

ダイヤモンド社
344890

初めてだってすぐわかる

SEGA® パソコン SC-3000 シリーズ BASIC入門

セガ・エンタープライゼス編
ダイヤモンド社

SEGA® パソコン SC-3000 シリーズ BASIC入門
セガ・エンタープライゼス編

ダイヤモンド社
344890

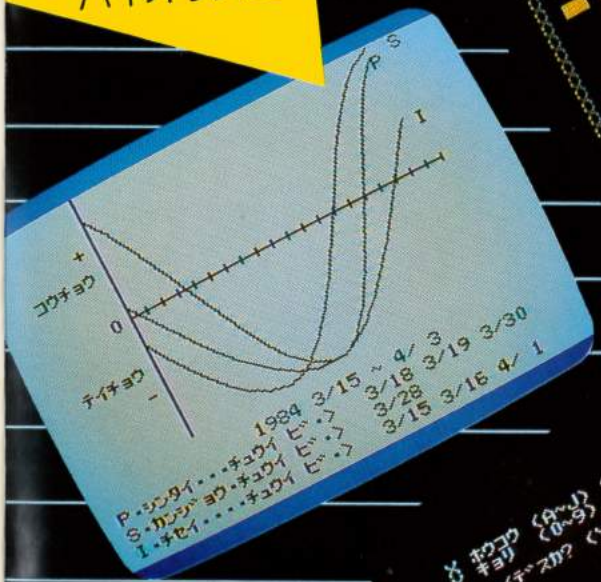
SC-3000シリーズ

SC-3000

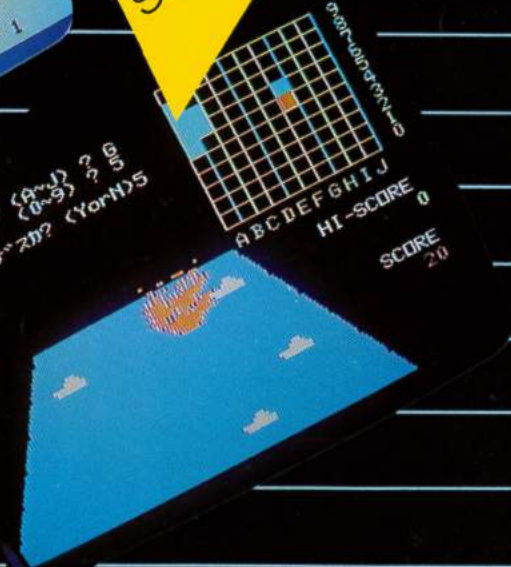
セガ・オリジナル
おもしろソフト紹介

ブロックくずし ゲーム

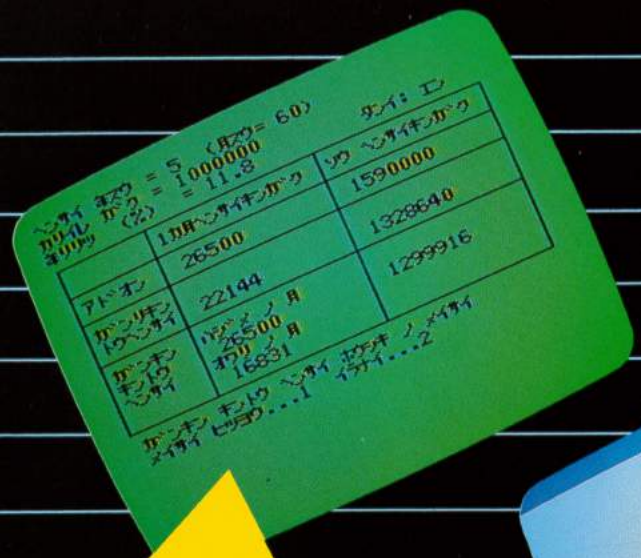
バイオリズム



SEA BATTLE



プログラム・リストは、巻末の
プログラム集にのっています。

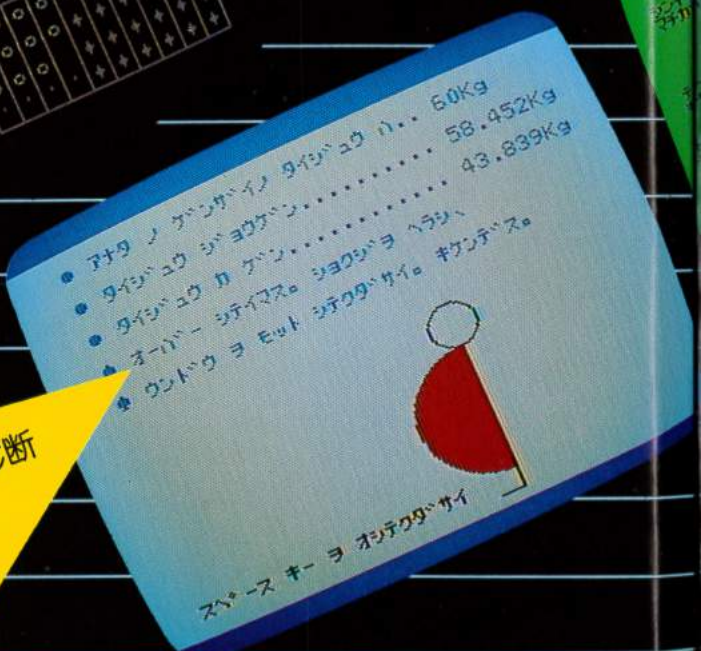


ローン計算

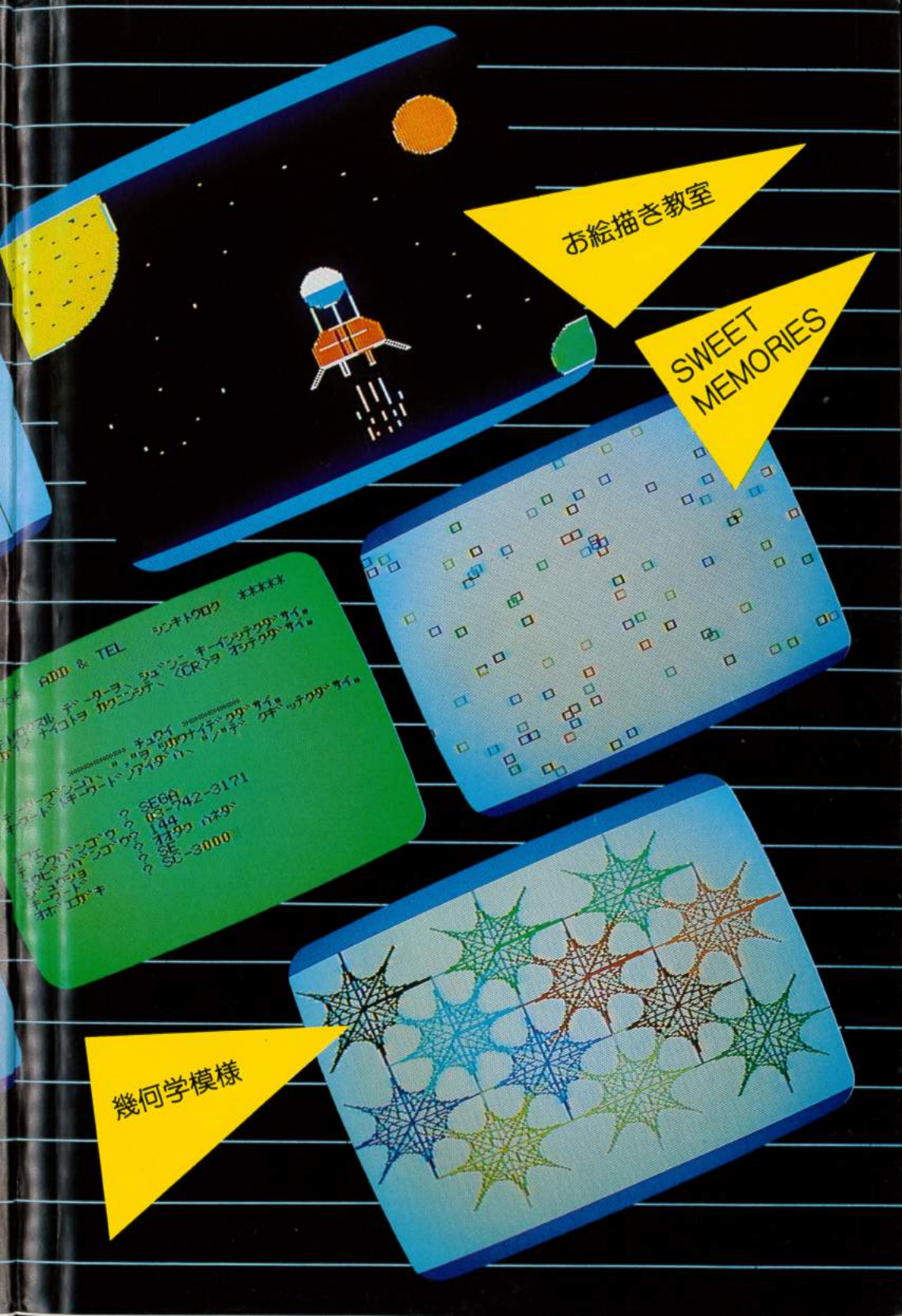
UFO撃墜ゲーム

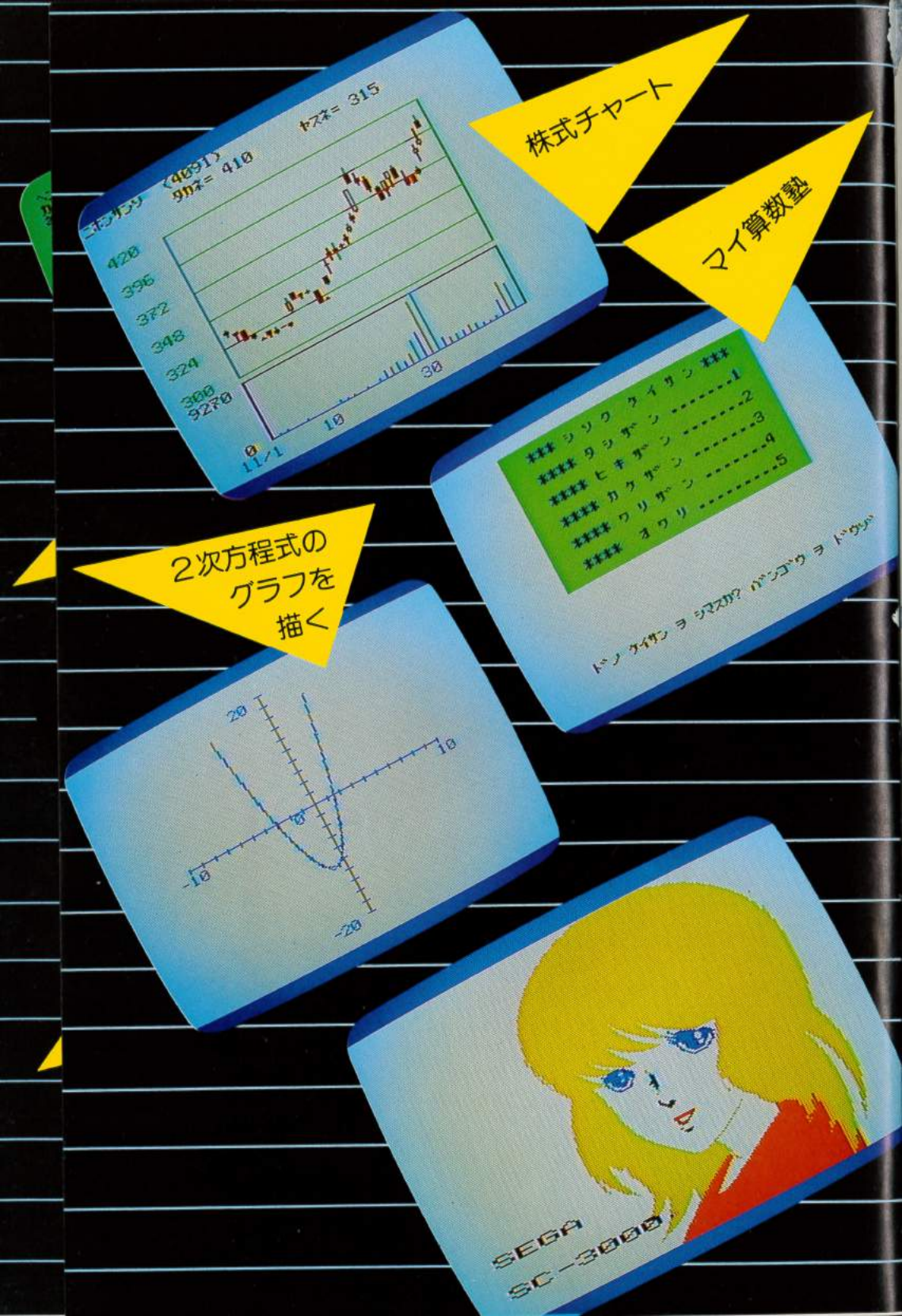
住所録

ナンバープール

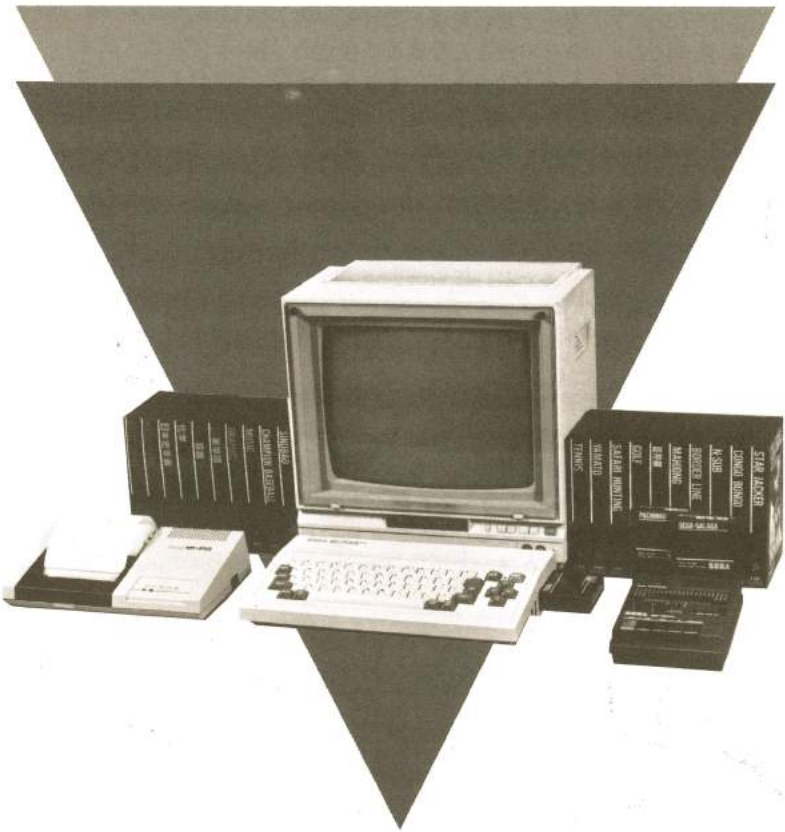


体重診断


お絵描き教室
SWEET MEMORIES
幾何学模様


株式チャート
マイ算数塾
2次方程式の
グラフを
描く
SEGA
SC-3000

SEGA
SC-3000 シリーズ
BASIC入門



目　次

（セガのパソコンにはSC-3000とSC-3000Hがありますが、本文中はSC-3000シリーズとします。）

第1章　コンピュータを動かす前に ―― 4

1. 電源スイッチを入れる前にこれだけはしておこう／4
2. キーボードに慣れよう／10
3. 英文字キーのもう一つの働き／18

第2章　プログラムを作らなくてもパソコンは動く ―― 21

1. プログラムを作るのはなぜ？／21
2. ダイレクト・モードなら電卓と同じ気軽さ／22
3. ダイレクト・モードはこんな場合に役立つ／24
4. ダイレクト・モードでもう少し遊ぼう／25
5. 計算に使う記号――演算子／26
6. 計算には優先順位がある／28

第3章　プログラム入門の入門 ―― 29

1. 簡単なプログラムを作る／29
2. プログラム・リストの訂正や消去はこうして……／35
3. 少し実用的(？)なプログラムを作る／38
4. 走り出したプログラムを止めるには／41

第4章　コンピュータらしい仕事をさせるプログラム ―― 43

1. 同じ仕事を繰り返すFOR～NEXT／43
2. 合格、不合格を振り分けられるIF～THEN／48
3. IF～THENを使って入力ミスを防ぐプログラムを作る／50
4. プログラムを短縮させるGOSUBとRETURN／52
5. メニューから品物を選ぶときはON～GOTO、ON～GOSUB／54

第5章　データ処理入門 ―― 56

1. プログラム上のデータを読み込む命令／56
2. 数値が多いときは「配列」を使う／58

第6章　プログラミングを便利にする命令 ―― 63

1. 不要な行を消す／63
2. 行番号を自動発生させる／64
3. 行番号を付け直す／65
4. プログラム・リストを見やすくする／66

第7章　カセット・テープにプログラムやデータを保存する ——67
1. プログラムの保存 ／67
2. プログラムの再生 ／68

第8章　コンピュータのおもしろさが倍増する「関数」 ——71
1. ゲームに欠かせない 乱数 ／71
2. 文字列に使われる関数 ／73
3. 時刻を表示 ／77
4. スペースを作る関数 ／79
5. どのキーが押されたか判断する関数 ／80
6. 残りのバイト数がわかる ／81

第9章　絵や図表を描くための命令 ——83
1. 画面を選択する ／83
2. 表示位置を決める ／83
3. 色をつける ／85
4. 線を引く ／87
5. 円を描く ／91
6. 区切られた部分を塗りつぶす ／93
7. 画面に点を打ったり消したりする ／94
8. 座標の原点とプラスの方向が決められる ／95
9. 文字や絵のパターンが作れる ／96
10. スプライト機能を使う ／99

第10章　その他コンピュータの機能をフルに活用するための命令 103
1. 音を出す ／103
2. 計算機能を高める関数 ／105
3. VRAM に直接作用する命令 ／106
4. プリンタ制御命令 ／109
5. ジョイスティックの命令 ／111

おもしろソフトプログラム集
1．ゲーム ／114
・ブロックくずしゲーム
・UFO撃墜ゲーム ／128
・SEA BATTLE
・ナンバー・プール
2．グラフィック
・お絵描き教室
・幾何学模様
3．音楽 ／133
・SWEET MEMORYS
4．学習 ／135
・マイ算数塾
・2次方程式グラフ
5．ホーム・ユース & ビジネス ／140
・体重診断
・バイオリズム
住所録
・株式チャート
・ローンの金利計算
付表 ／153
1. キャラクターコード
2. キャラクターセット
3. コマンド
4. ステートメント(命令)
5. 関数
6. エラーメッセージ
コーヒーブレイク
BASICとは ／9
変数について ／20
プログラムの1行とは ／40

第1章　コンピュータを動かす前に

1．電源スイッチを入れる前にこれだけはしておこう。

SC-3000シリーズを箱から出したとき、あなたはどんな気持ちでしたか。

この素晴しいパソコンを手にとって見たとき、きっと何か「未来的」なものを感じたことでしょう。

数々のキーが整然と並んでいて、本当に何でもできそうな感じがすると同時に、未来の国とか、将来の自分を想像したりしませんでしたか。SC-3000シリーズは「夢ふくらむ商品」と言えましょう。

SC-3000シリーズは業務用のコンピュータ・ゲームのメーカーとして知られるセガ・エンタープライゼス社が開発したものです。コンピュータ・ゲームにはかなり高度なハードウェアとソフトウェアの技術が必要なのですが、このSC-3000シリーズもそのような高度な技術を駆使して作り出されました。このように、SC-3000シリーズは優れた商品なので、永く使っていただくために次のことにはとくに気をつけて欲しいのです。

●コンピュータを使う場所やしまう場所に〝気くばり〟を

SC-3000シリーズは家庭用に作られたコンピュータですから、家庭内で使う限り、それほど気をつかわなくてよいのです。人間が生活できる環境下であれば、コンピュータも十分適応できるようになっています。

しかし、次のようないくつかの点については気をつけて下さい。

①温度……気温0℃以下のところではコンピュータを使う人はいないと思いますが、0℃以下のところに置いてあったコンピュータを取り出してきて、すぐに電源スイッチを入れますと、作動しな

いことがあります。このような時は少し時間を置いてからスイッチを入れて下さい。

また、陽が当たるなど高温になる場所には置かないようにしましょう。

コンピュータに使われている半導体（演算や記憶のために使われるコンピュータの主要部品）は一定の温度範囲の中で働きます。

②ホコリ…… SC-3000シリーズの場合はあまり気にすることはありませんが、コンピュータ一般としては、ホコリや煙は大敵です。コンピュータの会社で、社員全員が禁煙の会社もありました。

③振動……家庭内では機器に悪影響を及ぼすような振動はありませんが、コンピュータを落したりすることによって振動を与えたのと同じ影響を及ぼすことがあります。このようなことのないように、コンピュータの置き場所やしまう場所に気をつけましょう。

また、電源スイッチはあまりひんぱんにON-OFFをくり返さないようにしましょう。一度スイッチをOFFにしたら数秒間はONにしないように。

●本体に周辺機器を接続する

SC-3000シリーズ はキーボードのついたコンピュータ本体で、これだけでは何もできません。少くとも家庭用テレビと、ゲーム用のカートリッジかBASICのカートリッジが必要です。

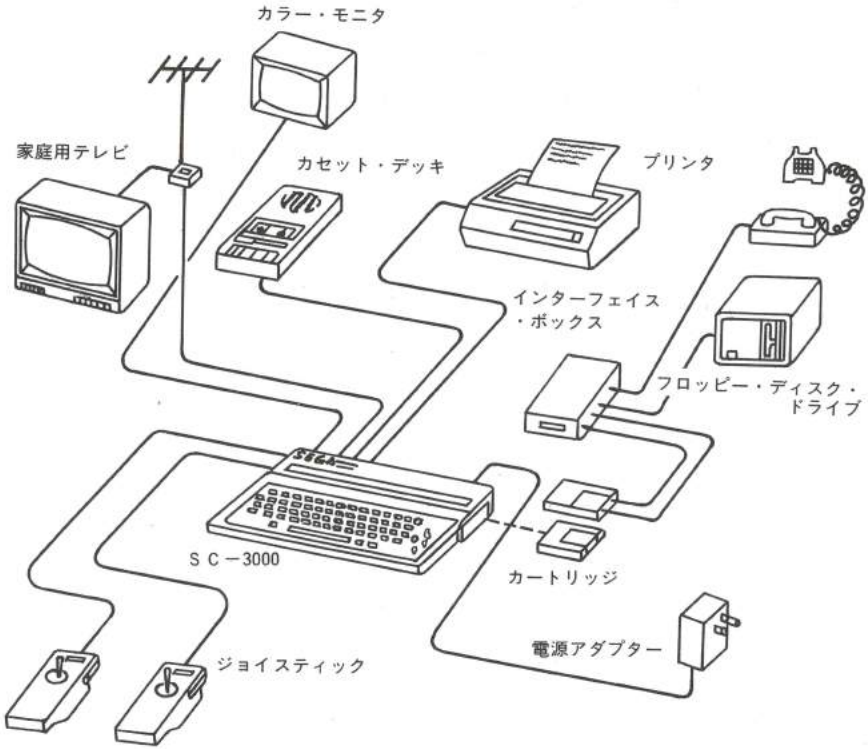
ゲーム用のカートリッジを使う場合は、本体にゲーム用カートリッジを差し込み、あとは家庭用のテレビと本体を線で結べばよいので簡単です。さらにジョイスティックを使うこともあります。

これに対して、自分でプログラムを組むという場合は、BASICのカートリッジが必要です。そして、作ったプログラムを紙に記録したり、カセット・テープに保存したいという場合はさらにプリンタとかカセット・デッキなど

の機器をつなぐ必要があります。

次の図は接続できるすべての機器をつないだ場合ですが、全部をつながなければならないというのではなく、目的に応じて必要なものだけをつなげばよいのです。

システム構成図



カセット・デッキやプリンタの電源アダプターは省略してあります。

●本体と周辺機器の説明

①本体……本体（SC-3000シリーズ）にはキーボードが付属しています。キーボードはプログラムやデータをコンピュータに入力する装置です。本体にはコンピュータの心臓部・頭脳部である CPU (Central Processing Unit——中央処理装置) という LSI (大規模集積回路) 化された半導体が使われています。

②カラー・モニタ……パソコンの周辺機器の中では欠かせないものです。データの入力は画面を見ながら行ないますし、パソコンが実行した仕事の結果を表示させたりします。またゲームをするにもカラー・モニタがなくては話になりません。モニタはディスプレイ装置または、CRT (Cathode Ray Tube ——陰極線管) ディスプレイなどと呼ばれることもあります。

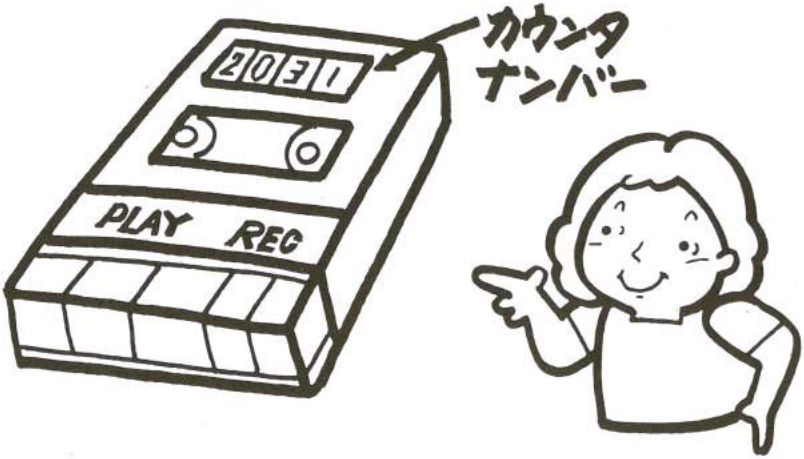
SC-3000シリーズには、家庭用カラーテレビをカラー・モニタとしてお使い下さい。その際は、付属のアンテナスイッチボックス (SS-50) を使います。また、SC-3000シリーズにはビデオ端子がありますから、ビデオ入力端子のついたテレビをお持ちの方は、別売のケーブルをお求めになり、よりよい画像をお楽しみ下さい。

③カセット・デッキ……作ったプログラムを記録・保存しておくために使います。市販のカセット・テープレコーダが使えますが、うまく記録できないものもあります。なるべく SC-3000シリーズ用のデッキ (SR-1000) をお使い下さい。

市販のカセット・テープレコーダの場合は、プログラムを記録するときは REC ボタンを、プログラムをパソコンへ呼び戻すときは PLAY ボタンを押します。

なお、本体（SC-3000シリーズ）の裏側のカセット端子の IN はカセット・デッキの OUT (市販品は EAR) へ、カセット端子の OUT はカセット・デッキの IN (市販品は MIC) へ接続して

下さい。



④ジョイスティック……ゲームをするときにキーボードの代りに使います。画面の中のUFOを撃ち落したり、カー・レースの車を左右に動かすときなど、動きのあるものにすばやく反応しなければならないゲームにはぜひ使いたい装置です。

⑤カラー・プロッタ・プリンタ（SP-400-J）……プログラム・リストを打ち出したり、プログラムの実行結果を印字するのに使います。カラーは4色でボールペン（専用）を使います。用紙は紙幅11.45センチのロール紙です。

⑥その他……フロッピー・ディスク・ドライブや電話を通じてパソコンどうしが通信できる音響カプラも準備中です。これらの機器をつなぐには、インターフェイス・ボックスを使います。

コーヒーブレイク

BASICとは

コンピュータに人間が命令を下すための言語の一つです。コンピュータはもともと0と1だけしか識別できないマシンで、コンピュータに命令するのもマシン内で最終的には0と1だけで行なわれています。しかし、0と1で人間が命令するのではたいへんで、コンピュータを使うのがイヤになってしまいます。こういったことで命令用の言語がいろいろ考えられてきて、パソコンではこのBASICという言語が今では一般的となりました。

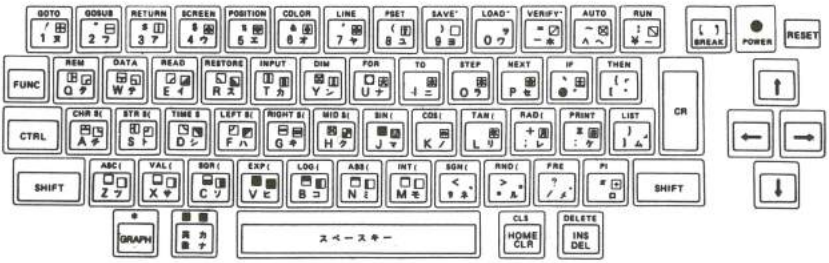
BASICはBeginner's All Purpose Symbolic Instruction Code（初心者向多用途言語）の略で、名前が示しているように、他のコンピュータ言語と比較しますと、たいへん理解しやすいものとなっています。パソコンが今日のように普及したのも、このBASICの発明によるところが大きいかも知れません。BASICは、アメリカのダートマス大学で開発されたと言われています。

2．キーボードに慣れよう

キーボードはコンピュータの入力装置のひとつであることは前にお話しました。キーボードはプログラムを作成するときや、プログラムを実行させるとき、また周辺機器を動かすときなど、私たちがコンピュータに仕事をさせようとするには、すべてキーボードを通して命令します。

コンピュータをあなたのよいパートナーにするには、まずキーボードをよく理解することです。

それではとにかくキーボードを見てみましょう。



キーボードを見ますと、キーの色が3種に分けられているのがわかります。それぞれの色のキーは次のように呼ぶことにします。

なお、SC-3000とSC-3000Hとではキーの色が違います。（　）の中はSC-3000Hのキーの色です。

黒色のキー（白）……文字・記号キー
灰色（グレイ）のキー（灰）……特殊キー
黄色のキー（赤）……リセット・キー

いろいろなキーを押す前に、まず「カーソル」について知っておきましょう。コンピュータの電源を入れますと、まず画面の左側に■が点滅（てんめつ）しています。これはカーソルといって、キーから入力される文字や記号が表示される位置を示しています。したがって、カーソルが点滅している時は「入力待ちの状態」ともいいます。

カーソルは次の3種類の形があります。

□……英数モード（電源を入れた時は英数モードです。）

■……カナ・モード

＊……グラフ・モード

上の3種のモードへの切り換え方は、次の〈文字・記号キー〉のところに書いてあります。

〈文字・記号キー〉

SC-3000シリーズのキーボードの英文字配列は英文タイプライターと同じ、カナ文字配列はJIS規格ですから、もっとも一般的な配列です。あなたがタイプライターを使ったことがあるなら、パソコンはとても身近かなものに思えるでしょう。しかし、たとえタイプライターを使ったことがなくても、パソコンの場合はとくに速くキーを打たなくてもよいし、文字の修正（しゅうせい）もとても楽ですから、キーボードにはすぐに慣れると思います。

まずキーを打ってみましょう。

[1 ヌ ノ 田] [2 フ ' 日] のキーを打って下さい。

写真のように、12と出ましたね。白いキー(SC-3000の場合は黒色)を単独で打つとキーの中の左側の文字や記号が画面に出ます。

数字のほか、ABC……とアルファベットを打ってみましょう。[;]や[:],[、]や[.]も打ってみましょう。

次は[SHIFT]（シフト）キー、[GRAPH]（グラフィック）キー、[カナ]キーを使って、キーの上側や下側などの文字や記号を画面に出してみましょう。

多くのキーは1つのキーの中にいろいろな文字や記号が書かれていますね。これは1つのキーでいろいろな文字や記号を書けるようにしたり、いろいろ

な働きをもたせるためです。SHIFTキーは1つのキーの中のいろいろな文字や働きを切り換える役目をします。

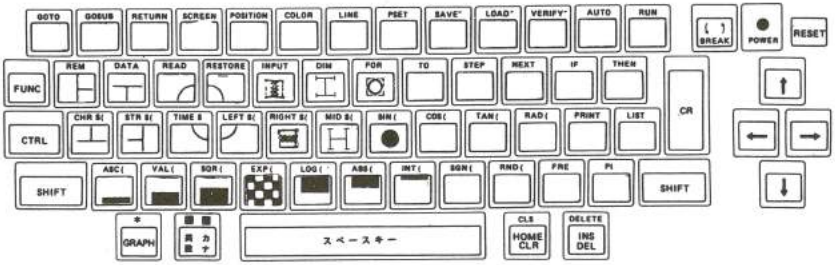
SHIFTキー

SHIFTキーは、SHIFTキーを押したまま他のキーを押して使います。SHIFTキーとアルファベット・キーをあわせて使いますと、小文字のアルファベットが画面に出ます。またSHIFT+数字・記号キーでは、数字・記号キーの左上の記号(SC-3000Hではキーの上側の記号、例えば1のキーでは！になります。

GRAPHキー

GRAPHキーは♣、♥、◢などのグラフィック記号を入力するためのものです。GRAPHキーを押しますと、カーソルは＊に変わります。そして、そのときのキーの働きは下図のようになります。

また、GRAPHモード（カーソルが＊の状態のとき）でSHIFTキーを押しながらグラフィック記号の2つあるキーを押しますと、左側のグラフィック記号が画面に出ます。

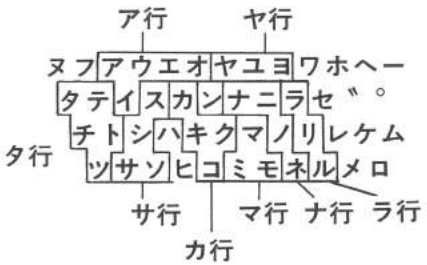


（SC-3000Hにはグラフィック記号がキーに印刷されていませんので、商品に付属しているシールをお貼り下さい。）

カナキー

カナキーはカタカナ文字を入力する場合に使います。カーソルは■に変ります。GRAPHキーと同じように、ふたたびカナキーを押すまでカナ・モードになっています。アイウエオヤユヨツの小文字はSHIFTキーを押しながら使います。

カタカナの配列は文字数も多いし、覚えるのに時間がかかりますが、案外かんたんに覚えられるかもしれません。キーボードの配列をよく見ますと次のようにある程度50音の行ごとにまとまっています。



カナ文字はア行、カ行の行ごとにほぼまとまっています。

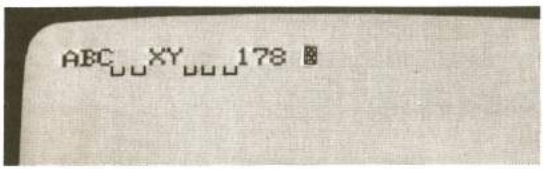
また、カナモードにおけるSHIFT+文字・記号キーでは、アイウエオヤユヨは小文字になり、ワがヲになります。また、「」、。・が使えます。

今までのキーを使って、色々な文字や記号を画面に映して下さい。適当なところでHOME CLRキーを押して、画面をキレイに消して下さい。そうするとまた新しく文字や記号を描けます。

〈特殊キー〉

キーボード上の灰（グレイ）色のキーが特殊キーです。しかし、特殊キーの中のGRAPHとカナキーについては〈文字・記号キー〉のところで触れましたので、ここでは省略（しょうりゃく）します。

SPキー



キーに何も書いてない細長いキーはスペース・キーです。この本ではスペース・キーはSPと書きます。スペース・キーは文字や記号の間をあける働きがあります。SPキーを1回押すと、カーソルが移動して1文字分の空間をあけます。本文中ではスペースを␣で表わします。

HOME CLRキー(ホーム／クリア)

このキーを押すと、画面上の文字や記号がすべて消え、カーソルが左上のポジション（ここの位置を「ホーム・ポジション」といいます。）に戻ります。画面上の文字、記号等がすべて不必要になった時に使って下さい。

なお、「ホーム／クリア」は画面に出ている文字や記号をクリアする(消す)だけで、後に出てくる「データのクリア」とか「プログラムのクリア」とは別で、ホーム／クリアを行なってもデータやプログラムはメモリーからは消えません。

SHIFTキーを押しながらHOME CLRを押すと、画面に表示されている文字は消えずにカーソルだけが、ホーム・ポジションに戻ります。左上の

方にある文字や記号を訂正するときに使うと便利です。

INS DEL キー（インサート／デリート）

画面に書いた文を訂正する時に使うキーです。このキーは、INS（挿入）と DEL(削除)の２つの役割をもっています。INS は SHIFT キーといっしょに押します。

DEL

例えば ABCD と入力するところを、ABCCD とキーをたたいてしまった場合、カーソルを消す文字の次の D の文字に移動します。そこで DEL を押すと直前の C が消えます。

① ABCC[D]　　D の文字にカーソルを合わせる（□はカーソルの位置です）。

② DEL を押す

③ ABC[D]　　C が１つ消える。

INS

例えば PQRST と入力するところを、PQRT とキーをたたいてしまった場合、カーソルを挿入したい個所のすぐ右側の文字、T のところへ移動します。そこで SHIFT+INS と押すとカーソルの点滅が速くなり R と T の間に文字や記号を挿入できる状態になります。

① PQR[T]　　T の文字にカーソルを合わせる（□はカーソルの位置です）。

② SHIFT+INS

③ S を押す。

④ PQRS[T]　　T の前に S が入る。

なお、カーソルが速く点滅している間は何文字でも挿入できます。そしてインサート・モード（インサートできる状態にあること）から抜け出すには、次のいずれかの操作をします。

a. CR キーを押す

（ただし、これまでの学習の範囲内の操作だけでは ？ Syntax error

の表示がでます。)

b. カーソル・キーを押す。

c. SHIFT+INSキーを押す。

CRキー（キャリッジ・リターン）

「キャリッジ・リターン」という言葉はタイプライタの用語で「改行」を意味しますが、実際に、CRキーを押すと、画面は改行した状態になりますね。コンピュータではこのとき「文字や記号をメモリーに記憶させる」などの働きを同時にします。

何か文字（ASIF135Gなど、頭に数字のない文字列）を入力してCRキーを押して下さい。?Syntax error（シンタックス・エラー）と出ましたね。コンピュータに入力する場合、一定のきまりで入力しなければなりません。これをコンピュータの文法といいます。文法が間違っているとエラーになります。文法といっても、私たち日本人は日本語の文法などあまり意識しないで日本語を話しています。どこの国の人も文法などふだんはあまり気にしていないのと同じように、BASICも慣れてくればあまり気にしなくてすむようになるのです。なぜなら、BASICに使われている言葉は、日本語や英語とは比べものにならないくらい少ない単語や熟語でできているのですから。それに、同じことをやらせるにも、1つの決まった方法しかないのではなく、プログラムを作る人によって、いく通りもの方法が考えられることが多く、表現方法にも柔軟性があります。BASICは世界のどこか小さな小さな国の言語を覚えるつもりで勉強しましょう。

FUNCキー（ファンクション）

プログラムを組むには、INPUTとかPRINTなどの命令を使います。こういったBASICの命令をキーボードから入力するには、INPUTとかPRINTと1文字ずつ入力してもよいのですが、これらの手間を省くために、1つのキーで1つの命令を入力できるようにしたのがこのFUNCキーです。

たとえば、英文字のTのキーはINPUTの代用をしますから、INPUT

と打ちたいときは、FUNCキーを押したまま、Tキーを打って下さい。

なお、SC-3000Hの場合は、命令の文字はキーには書いてありませんから、付属品のシールをキーの側面に貼って下さい。

BREAKキー（画面切替え／ブレイク）

次の2つの機能をもっているキーです。

①**ブレイク**……プログラムの実行中にプログラムの実行を停止させたい時、または長いプログラム・リストの訂正個所を見つけるために、リストを画面に出し、画面の中でせり上っていくリストを途中で停止させたい時に（停止はSPキーで）押します。

②（**画面の切替え**）……SC-3000やSC-3000Hはテキスト画面とグラフィック画面の2つの画面をもっています。SHIFT＋でテキスト画面→グラフィック画面、グラフィック画面→テキスト画面と切り替えます。

（テキスト画面、グラフィック画面については、第9章の2をお読み下さい。）

RESETキー（リセット）

キーボード上の右上端にある、黄色い（SC-3000Hは赤）キーです。

プログラム実行中、または画面に異常が出た場合、RESETキーを押しますと2～3秒後に電源を入れた時と同じ画面に戻ります。ただし、使用していないメモリーの大きさの表示は、入力されているプログラムやデータの大きさによって違います。

（このキーを押しても、入力されているプログラムやデータは消えません。）

3．英文字キーのもう一つの働き

この項はプログラムを作ることにある程度慣れてからお読み下さい。アルファベット・キーにはこんな機能もあるのだナ、ということを頭の中に入れておいて下さい。

CTRLキー（コントロール）

CTRLキーを押したまま、アルファベット・キー（全部ではありません）を押すと、下表のような動作が行なわれます。

キー操作	PRINT CHR$（値）;	機　　能
CTRL + A	PRINT CHR$（1）;	NULL 文字なし
C	——	BREAK プログラムの実行中止
E	5	カーソル以降の文字をクリア
G	7	BELL ピッと音を出す
H	8	DEL 文字をデリート
I	9	HT 水平 TAB
J	10	LF ラインフィード
K	11	HM カーソルを左上端に戻す
L	12	CLR 画面のクリア
M	13	CR キャリッジリターン
N	14	カナ ↔ 英数字切りかえ
O	15	{ } 画面をテキスト ↔ グラフィック切りかえ
P	16	標準文字サイズ
Q	17	横2倍文字サイズ (SCREEN 2)
R	18	INS インサート
S	19	キー入力（A～Z）シフトなし大文字
T	20	〃　（a～z）シフトなし小文字
U	21	ラインをクリアしカーソルを左端に戻す。
V	22	ノーマルモード
W	23	GRAPH キー入力グラフモード ↔ 英字切換
X	24	クリック音の ON ↔ OFF 切換
	28	⇨ カーソル移動
	29	⇦　〃
	30	⇧　〃
	31	⇩　〃

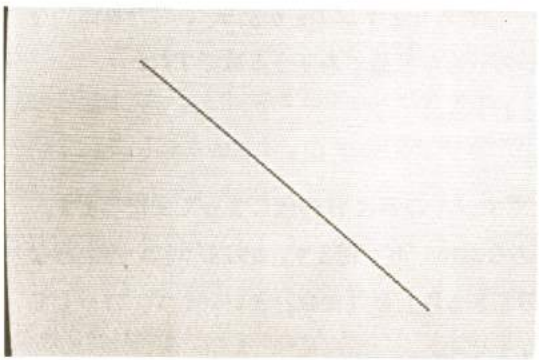
プログラム中でコントロール・コードを使う場合は、PRINT CHR$(数値)

で入力して下さい。この場合の PRINT は画面に CHR$ (数値) を表示する意味ではありません。

以上でキーの説明は全部終わりました。キーを使って次のプログラムを打ち込んでみましょう。

```
10  SCREEN 2, 2 : CLS CR
20  LINE (50, 50) - (150, 150), 5 CR
      ↑線を引く  ↑x座標 ↑y座標  ↑x座標 ↑y座標  ↑色
30  GOTO 30
RUN CR
↑プログラムを実行させる命令です。
```

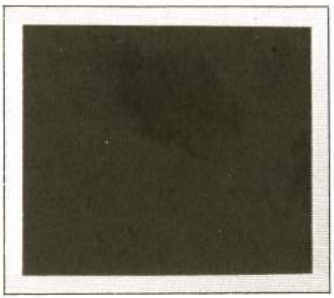
写真のような線が引かれたと思います。



こんどは色番号の後に, B を追加して下さい。(カーソルを 5 の後にもって行きます。,B を打ち込んだ後 CR キーを押すのを忘れないように。)

そして、カーソルを下の方の何も書いてない行へもって行き、RUN CR と操作します。画面には四角が描かれたと思います。

次に、今追加した ,B のあとにさらに ,F を追加して下さい。これで RUN させると、四角の中にうすい青色が塗られたと思います。



コーヒーブレイク

変数について

「変数」は「数」の一種でありますが2とか3.5といった具体的な数ではありません。いろいろな数値であり得る数ですね。

次の式で考えてみましょう。

A＝B×5

AやBは「変数」です。Aの値はBの値によって変化します。またBの値もいろいろな数値であり得ます。このようにいろいろな数値に「変わることのできる数」を「変数」といいます。けっして「ヘンな数」ではありません。

さて、SC—3000で使える変数の型は次のように決まっています。

先頭の1文字が英字、第2文字が英字または数字です。3文字以上にしてもかまいませんが、第3文字以降の文字は無視されます。ただし、無視されるからといって、記号やカナを入れるとSyntax error（文法的な誤り）となります。

〈変数の例〉

①英文字1字……A　B　P　Q　Z　など

②英文字2字……AA　AB　CT　QR　ZZ　など

③英文字1字＋数字1字　A1　F3　S0　Z9　など

第2章　プログラムを作らなくても パソコンは動く

1．プログラムを作るのはなぜ？

プログラムとかプログラミングというのは一体どういうことなのでしょうか。「プログラム(Program)」を英和辞典で引きますと、計画表とか段どり、手順という訳が見当たります。ですから「コンピュータにプログラムを記憶させる」と言えば、「コンピュータに仕事の手順を覚え込ませる」ということになり、また「コンピュータにプログラムを実行させる」と言えば、「コンピュータに手順どおりの仕事をさせる」ことを意味します。なお、「プログラミング」は人間がプログラムを作ることを言います。そして、プログラムを人間の目で見えるようにしたものをプログラム・リストといいます。プログラム・リストはこの本の文中や巻末にもいくつか載っていますから、どんなものであるかは分ると思います。

では、プログラムを理解するために次の平均値の計算をやってみましょう。

(65＋80＋37＋43＋72＋12＋53)÷7＝　?

上の計算の電卓でするならば、まず65＋80＋…＋53を計算し、そしてその結果を7で割ります。

こんどは、この計算をプログラムを使ってやってみることにしましょう。

```
10  FOR I= 1  TO 7
20  INPUT   A
30  T=T+A
40  NEXT   I
50  PRINT   T／7
```

上のプログラムの意味はまだわからなくていいです。これで計算すると、パソコンのキーボードから、65、80、…、53と数値を入力するだけで答が出ます。＋、÷、＝などの記号は使いません。

このように、プログラムを使って仕事をすると手間が省けます。上の例は簡単な計算ですが、もっと複雑な計算式の場合や、ひんぱんに同じ式の計算をする場合は、電卓とは比べものにならないくらい速くできます。

またパソコンはただ計算をするだけでなく、論理判断も高速で行ないます。上の平均値の計算で、65以上なら合格、65未満なら不合格と決め、合格、不合格の文字を出すようにしたければプログラムでそのようにもできます。コンピュータは「電子計算機」と訳されるように、計算をする機械ですが、計算といっても四則演算のような計算だけでなく、数値の大小の比較や条件判断をしますから、これらをうまく組み合せていろいろなことができるのです。パソコンもコンピュータの一種ですが、今ではパソコンは電子計算機と思うよりは、テレビゲームと思う方のほうが多いくらいで、「パソコンで計算もできる」というと、びっくりする方もいます。

パソコンに限らず、各種のテレビゲームもプログラムがあるからゲームを楽しめるのです。パソコンにゲームのルールや手順を憶え込ませればゲーム・マシンになります。このルールや手順をコンピュータの理解できる言葉で書いたものがプログラムですね。

プログラムというものがどんなものか、だいたいのところは分かっていただけたと思いますが、いかがですか。それでは次へ進んで、実際にパソコンを動かしてみましょう。

2．ダイレクト・モードなら電卓と同じ気軽さ

コンピュータに計算をさせるには、普通は前もって計算の手順であるプログラムを記憶させます。しかし、簡単な計算をするときは、いちいちプログラムをコンピュータに記憶させないで、直接、計算にとりかかるよう命令することができます。このことを「ダイレクト・モード（直接命令）で計算を

させる」といいます。なお、直接命令といってもプログラムを組むときと別の命令があるわけではありません。プログラムを組む場合に用いるいろいろな命令をそのまま使います。

ダイレクト・モードで計算をする場合と、プログラムを組んで計算をする場合とでは、次のようにキー操作の手順が違います。

3×5の計算をする場合で比較してみましょう

●ダイレクト・モードで計算をする場合

[P][R][I][N][T] [3][*][5] [CR]

15　　かけ算の記号です

●プログラムを組んで計算をする場合

[1][0] [I][N][P][U][T] [A] [CR]
[2][0] [I][N][P][U][T] [B] [CR]
[3][0] [P][R][I][N][T] [A][*][B] [CR]
[4][0] [E][N][D] [CR]

この部分がプログラムです。

[R][U][N] はプログラムを実行する命令です。

? [3] [CR]　?は入力を待っている記号です。
? [5] [CR]

15

なお、ダイレクト・モードで使う [CR] とプログラムを作るときに使う [CR] とは意味が異なります。ダイレクト・モードの [CR] は PRINT 文の実行を意味しますが、プログラムの [CR] はプログラムをコンピュータのメモリーに記憶させることを意味します。

ダイレクト・モードでは必ず PRINT 命令を使いますが、PRINT 命令を使いますと、「PRINT」の次に来る式を計算して、画面に表示します。

簡単な計算をダイレクト・モードでやってみましょう。

① PRINT　5－8 [CR]
　－3
② PRINT　(7＋2)＊4 [CR]
　36

③ PRINT　2＊PI／6 CR

1.0471975512

／ ↑ 割り算の記号です

π（円周率 3.14……）を表わしています。

PRINT 命令はまた、文字を画面に表示することにも使えます。

PRINT "HANAKO SAITO" CR

と打って下さい。画面に次のように出たと思います。

上の PRINT に続く "（ダブル・クォーテーション）と" の間の文字のことを「文字列」といいます。文字列には、アルファベットの他、数字、カナ、記号が使えます。

なお、文字列の中に"を使いたいときは、"を２個並べて下さい。（２個で１個の "を表わします）

PRINT "PASOCON WA" "SEGA" "DA" CR を打ち込んで下さい。

PASOCON WA"SEGA" DA

と出ましたね。

3．ダイレクト・モードはこんな場合に役立つ

ダイレクト・モードは簡単な計算をする時だけでなく、プログラムの一部が間違っていないかどうかを確かめる時にも使います。プログラムを組んでいる時に、一時中断して、行番号を書かずに、ダイレクト・モードで命令文を作り、実行してみることができます。

例えば、プログラムの中に

X＝A÷B

という文があったとします。÷という記号が割り算の式の中で使えるのかどうか分からないときに、A÷Bを実行してみればよいのです。この場合、AとBにわかりやすい数値を代入して計算します。

```
PRINT  12÷ 4
          ↑
          └グラフィック記号の÷
```

残念ながら、上の文章を実行させようとしますと、? Syntax error（文法上の誤り）と出て、書き方に誤りがあることがわかります。そこで÷の記号はやめて、／（スラッシュ）に書き替えましょう。

こんどは3と表示されたと思います。÷の記号は使えないことが分りました。

4．ダイレクト・モードでもう少し遊ぼう

ディスプレイ画面にABC……とAからZまでのアルファベットを並べようとしたらどうしますか。

PRINT〝ABCDEFG……Z〞 CR

というように書きますか？　めんどうですね。次の文章を入力して下さい。

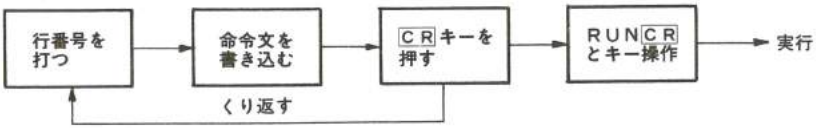
FOR　I＝65　TO　90：PRINT CHR$ (I)；：NEXT CR

A、B、Cなどの文字がプログラムの中に入っていないのに表示されました。不思議ですね。しかし、この本を読んでいくうちに、理解できるようになりますから安心して下さい。

ではこのへんで直接命令でコンピュータに仕事をさせる場合と、プログラムを組んで仕事をさせる場合の違いを整理してみましょう。

●直接命令で仕事をさせる場合

●プログラムを組んでから仕事をさせる場合



5．計算に使う記号——演算子

コンピュータに使われる計算のための記号は、私たちがふだん使っているものと違っているものがあります。それらは英文タイプライターから来ているものです。英文タイプライターはもともと英文手紙を書くためのものですから、計算をするための記号はありませんでした。そこでかけ算に使う×の代りに＊（アスタリスク——星印）、わり算の÷の代りに／（スラッシュ）などが用いられるようになりました。

計算に使う記号のことを演算子と呼びます。次は演算子の一覧表です。

	記号	使用できる場所		内　　容	優先順位
		数値式	文字式		
算術演算	∧	○	×	べき乗　（0∧0は1）	1
	＋	○	×	符号　＋	2
	－	○	×	符号　－	
	＊	○	×	乗算	3
	/	○	×	除算	
	MOD	○	×	余り	4
	＋	○	○	加算　（文字式の場合は文字の結合）	5
	－	○	×	減算	

- カッコ（　）が最も優先度（ゆうせんど）が高い。
  同じ優先順位の演算子が2つ以上使われた時は、左側の演算を優先します。
- 加算記号　＋　を文字式に用いた時は、結合を表わします。
  (例)　"AB"＋"C" ⇨ "ABC"

● 論理演算は2進で行なっています。

● 算術演算は10進12けた計算、11けた表示です。

例えば0.01きざみの値でもけたおちがありません。

演算子と違いますが、演算子と同じように数式に用いられる( )カッコがあります。また、これ以外に数値の大きさを比べるための比較演算子というのがあります（51 ページ参照)。

ダイレクト・モードで四則演算をやってみましょう。

たし算

```
? 3＋5 CR
 8
Ready
```

注1　? は PRINT の略号です。
? は SHIFT キーを押しながら [?/] を押します。

注2　3＋5 の次の行の 8 の前 1 字分が空いているのは、＋記号が省略されているからです。

ひき算

```
? 2－8 CR
－6
Ready
```

かけ算

```
? 7＊6 CR
 42
Ready
```

わり算

```
PRINT 10／3
 3.3333333333      (小数点以下10桁)
```

また、「余り」を知りたいときは PRINT 10 MOD 3 です。

PRINT は FUNC キーを押しながら [*明] を押しても結構です。

大きな数や小さい数は次のように表示されます。

？１９８４０００００＊１００００００００CR

１．９８４Ｅ＋１６

上の１．９８４Ｅ＋１６は$1,984\times10^{16}$を表わします。

？３／1000CR

３Ｅ－０３

これは3×10^{-3}を表わします。こういった書き表わし方は、指数的(しすうてき)表記法といいます。

６．計算には優先順位がある。

２つ以上の演算子の入った数式の計算は先頭から行なわれるでしょうか。

？３＋６＊５CR

３３

となり、４５ではありません。この場合は６＊５のかけ算から先に行なわれます。優先順位は、先の演算子表を参照して下さい。

上の式で３＋６を先に計算したいときは、

？(３＋６)＊５CR

とします。かけ算、わり算は自動的に優先するようになっていますから、たし算、ひき算の方を優先させたいときは( )カッコを使います。

次の式はちょっと複雑です。カッコが何重にも使われていますが、大カッコ、中カッコはありません。コンピュータでは小カッコしかありません。

？４＊((59－３)／(５＋２))　④ ① ③ ② ←優先順位

３２

上の式の計算は優先順位の番号の①→④の順に行なわれます。優先順位が同じであれば、左側から先に計算されます。

カッコを使う場合、左カッコと右カッコが同数でないとエラーになります。カッコをたくさん使うときは、よく注意しましょう。

第3章　プログラム入門の入門

1．簡単なプログラムを作る

コンピュータのプログラムを作る、というと何か非常にむずかしいことのように思えるかも知れません。その理由としては、① BASIC というプログラムを作るために使われる言語が英語であること、②いろいろな命令を完全に理解しなければプログラムを作れないと思い込んでいることの２つが挙げられるのではないでしょうか。

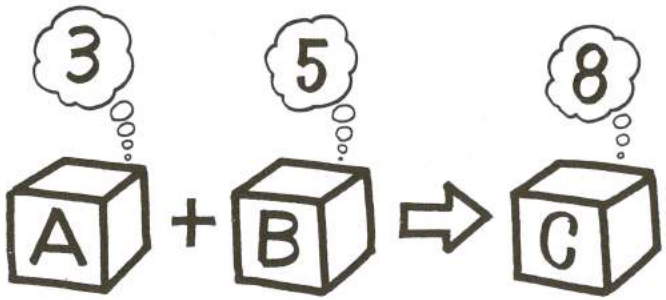
たしかに BASIC は英語です。しかし、プログラムを作っている多くの人が、BASIC は英語であると意識してはいないと思います。X＝A＋B という公式を英語だと思う人はいないのと同じように、INPUT とか、PRINT といった命令はプログラムを作る人にとっては記号ぐらいにしか思われません。少し慣れてくれば、あなたもきっと BASIC は英語だなんて思わなくなるでしょう。

また、プログラムを作るための命令はたくさんありますが、これらの全部を理解しなくても、プログラムは作れます。第一、どんなプログラムでも、命令のすべてを使っているプログラムなんてないと思います。わずか５～６種の命令だけでもプログラムは作れます。この章ではもっとも分りやすい命令だけでプログラムを作る練習をしましょう。

RUN、LET、変数、PRINT

次のプログラムを見て下さい。

```
10  LET  A=3
20  LET  B=5
30  LET  C=A+B
40  PRINT  C
50  END
```



プログラムの1行は、行番号とステートメント（文）とから成り立っています。各行の頭にある10、20、……の数字が行番号（ライン・ナンバー）です。行番号に続いて文を書きます。プログラムは1行だけでも成り立ちますが、ふつうはたくさんの行数からできています。

では、上のプログラムをキーボードから入力してみましょう。

HOME CLR	1	Ø	SP	L	E	T	SP	A	=	3	CR
	2	Ø	SP	L	E	T	SP	B	=	5	CR
	⋮										

と入力していきます。SPはスペース・キーです。SPキーは打たなくてもプログラムとしては有効ですが、プログラムを見やすくするために使います。また、LETは省略可能な命令です。命令自体が省略可能なのは、このLETぐらいです。

全部入力し終わると、画面は次のようになります。

```
10 LET A=3
20 LET B=5
30 LET C=A+B
40 PRINT C
50 END
```

そこで CLS キーを押して、いったん画面をきれいにしてから、
RUN CR と操作して下さい。下の図のようになるはずです。

```
RUN
 8
Ready
```

RUN はプログラムに書かれていることをコンピュータに実行させる命令で

す。RUNは「ルン」ではありません。「ラン」と発音します。なおRUNはプログラムの中では使えません。このようにプログラムの中では使えない命令のことを「コマンド」と言います。

もう一度前ページのプログラムを見て下さい。このプログラムは3と5をたす、たし算のプログラムです。この中に使われている命令のLET、PRINTについてお話ししましょう。

LETは英語では「～させる」という意味ですが、BASICでは上のプログラムに見られるような数式（A＝3のような）の前につける命令となります。

LETで始まる文は代入文と呼ばれます。この代入文の中に使われる記号（＝）はイコール（等号）ではありませんから注意して下さい。上のプログラムでA＝3は、Aという変数に3を入れるということを意味します。C＝A＋Bは、Cという変数にAとBとをたした結果を入れることを意味します。次のような数式の中の＝はどうでしょうか。

A＝A＋B

B＝B＋1

いずれも＝がイコールであったならば、算数や数学ではあり得ないことです。しかし、BASICの代入文ではこういった式が成り立つのです。そして、この形はプログラムの中にかなりよく用いられます。

A＝A＋B

A＋Bを計算してその答えをAに入れます。したがってAの値はA＋Bを計算する前と後では異なります。

話を前に戻してこんどは、40行を次のように変えてみましょう。

40　PRINT "A＋B="；C

こうすることによって、単に答えだけでなく、どのような計算がされたのかがわかります。

PRINT文の中に""ではさんだ文章（これをプロンプトといいます）を入れますと、わかりやすい、親切なプログラムだということになります。

さらにプロンプトを次のように変えてみましょう。

40 PRINT"コタエ␣ハ␣" ；3＋5；"␣エン"

RUNさせますと、次のようになります。

```
RUN
コタエ ハ 8 エン
Ready
```

文字変数

単に「変数」といった場合は数値の場合を言いますが、「文字変数」といって、文字を扱う変数もあります。

前のプログラムを文字変数を使って書き替えてみましょう。

5 A$＝"コタエ␣ハ␣"：B$＝"␣エン"

40 PRINT A$；C；B$

これを実行すると前と全く同じに、

コタエ ハ 8 エン

となります。5行でA$という文字変数に"コタエ ハ "という文字列とB$という文字変数に"エン"という文字列が入りました。これを40行のPRINT命令でA$とB$の中身を読み出しています。

文字変数は$(ダラー) マークをつけます。

例 A$、B$、AB$、XY$、D1$

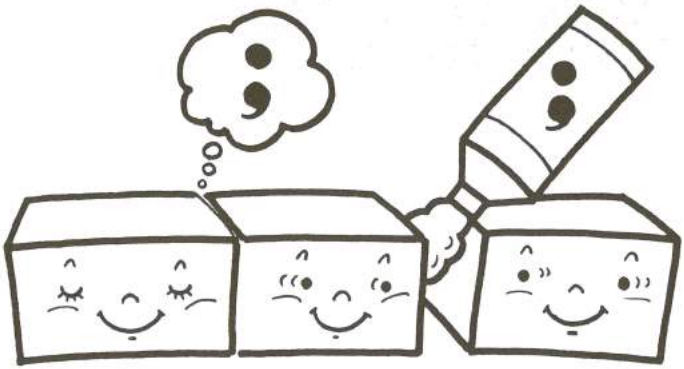
そして、文字変数に文字を代入するには〝(ダブル・クォーテーション)を使います。

また、2つ以上の文字変数はつなぐこともできます。

```
10 A$="I "
20 B$="AM "
30 C$="A BOY"
40 D$="."
50 E$="?"
60 PRINT A$+B$+C$+D$
```

```
RUN
I AM A BOY。
Ready
```

文字変数をつなぐには＋記号または；（セミコロン）を使います。



```
10 A$="I "
20 B$="AM "
30 C$="A BOY"
40 D$="。"
50 E$="?"
60 PRINT B$;A$;C$;E$
```

```
RUN
AM I A BOY?
```

PRINT命令は下図のリストの60行のように、PRINTだけで終っている場合は、「改行」を意味します。画面に行をつめていろいろな数値や文字を表示しますと見にくい場合があります。そのようなときは、空白の行を作るために、PRINTを使います。2行あけるときはPRINT：PRINTとします。

```
10 LET A=5
20 LET B=12
30 LET X=A+B
40 LET Y=A*B
50 PRINT X
60 PRINT
```

```
70 PRINT Y
80 END
```

```
RUN
 17

 60
Ready
```

2．プログラム・リストの訂正や消去はこうして‥‥

プログラムは初めから完全なものはできません。Syntax error（文法上の誤り）をなくすことは当然のことですが、より使いやすくするとか、いろいろな機能をもたせるなどのために、プログラムを訂正することがあります。訂正のしかたが分からないでプログラムを作っていると不安になります。本格的なプログラムを作る前に、訂正のしかたをよく覚えましょう。また、続けて違うプログラムを入力するときのために、メモリーを消去する方法を覚えましょう。

LIST

完成した、あるいは未完成のプログラムをディスプレイ画面に映したいときは、LIST CR と操作します。この LIST も RUN 同様、プログラムの中では使えません。

LIST CR ではコンピュータに記憶されているプログラムの最初の行から最後の行まですべて表示されます。また、LIST に続く文の書き方で、次のような表示のしかたがあります。

LIST　行番号	1行のみ表示
LIST　行番号—行番号	行番号から行番号までを表示
LIST　行番号—	行番号からうしろのプログラムを表示
LIST　—行番号	プログラムの先頭から行番号までを表示

—は , キーを使っても同じです。(—は -=ﾎ キーのマイナス記号です。)

LIST で表示されたプログラムは、カーソルを移動して書き替えられます。

```
LIST

 5 A$="ｺﾀｴ ﾊ":B$="ｴﾝ"
 10 PRINT "A$;3+5;B$
■
```

5行のコタエ　ハの文字をキンガクに直したいときは、カーソルを下図のように、コの文字のあったところに移動します。

```
LIST

 5 A$="■ﾀｴ ﾊ":B$="ｴﾝ"
 10 PRINT A$;3+5;B$
```

そして、そのままキンガクとキーを打ちます。

```
LIST

 5 A$="ｷﾝｶﾞｸ■"
 10 PRINT A$;3+5;B$
```

上の図のようになったと思います。ここで[CR]を押すと、訂正は終了です。各行の訂正が終るごとに[CR]キーを押すのを忘れないで下さい。CRキーを押し忘れますと、コンピュータ本体内のメモリー内のプログラムの訂正は行なわれません。

プログラムの行数が多いとき、LIST[CR]で表示をすると、画面の中でリストがいっぱいになると、リストがせり上っていきます。このような時、リストを止めてプログラムを見たいときは、[SP]キーを押します。続けて[SP]キーを押しますと、またリスト表示が再開されます。これは何回でも、リスト表示が終わるまで繰り返すことができます。

リストを止めて、プログラムを訂正するときは[BREAK]キーを押します。するとカーソルが点滅しますので、カーソルを移動して訂正して下さい。

```
LIST
10 A$="I "
20 B$="AM "
30 C$="A BOY"
40 D$="."
50 E$="?"
Break
```

50行まで表示されたところで SP キーを押した

訂正したあとで再び LIST を動かすときは、LIST 行番号— と改めて操作して下さい。

NEW

新しいプログラムを入力する場合、前に入力したプログラムがコンピュータ内のメモリに残っていると、新しく入力したプログラムと混じってプログラムが正常に動かない場合があります。電源を入れた直後にプログラムを組むのであれば、このような心配はありませんが、次から次とプログラムを作るときは、前に入力したプログラムを全部消してから次のプログラムを入力しましょう。

プログラムを全部一度に消すのは、NEW CR でよいのです。簡単ですから、せっかくのプログラムを間違えて消すことのないように気をつけて下さい。

```
NEW
LIST
```

NEW CR としたあと、LIST CR で、カーソルだけしか映りません。プログラムはキレイに消えました。

3．少し実用的(？)なプログラムを作る

いままでに覚えた命令にCLS、INPUT、GOTOの3つの命令を加えてたし算を行なうためのプログラムを作ってみましょう。この章のはじめに作ったプログラムを少し変えて、実際の計算に役立つようにしましょう。

CLS、INPUT、GOTO

次のプログラムを入力してみましょう。

```
10 CLS
20 INPUT A
30 INPUT B
40 C=A+B
50 PRINT C
60 GOTO 20
```

10行のＣＬＳは画面をクリアする命令です。

20行のINPUT Aは「Aという変数に数値を入力せよ」という意味です。INPUT文が実行されますと、画面に？マークが現われ、キーボードから数値を入力することになります。30行も同様です。

入力した数と答が画面に出たと思います。

なお、上のプログラムで20行と30行をいっしょにして、

```
20  INPUT A, B
```

としてもよいのですが、その時の画面は次のように、Bの入力待ちのカーソルの前に？マークが2つ付きます。入力する変数が3つ以上になっても？マークは2つです。

```
RUN
? 5
??
```

INPUT 文は、PRINT 文と同じように""ではさまれたメッセージ文を使えます。

プログラムの流れが、GOTO 文に出会うと、GOTO 文で指定された行番に飛びます。

三角形の面積を求めるプログラムを作ってみましょう。三角形の面積は、底辺の長さ×高さ÷2ですね。

プロンプト文を入れることによって、何を入力したらよいかがわかり、たいへん便利です。

```
10 INPUT"テイヘン=";A
20 INPUT"タカサ=";B
30 M=A*B/2
40 PRINT"メンセキ=";M;"ヘイホウ メートル"
50 END
```

```
RUN
テイヘン=25
タカサ=4
メンセキ= 50ヘイホウ メートル
```

END

上のプログラムの最終行である50行に END という文が出てきました。END はコンピュータに仕事を終らせる命令です。END はプログラム・リストの中でいちばんおしまいの行にあるとは限りませんが、上のような簡単なプログラムではたいていいちばん最後の行にあります。そしてこのように、いちばん最後の行にある END は書かなくても支障はありません。

BREAK

前のプログラムで、60行に GOTO 文があります。GOTO 20となっていますから、PRINT Cを実行したらまた20行の INPUT 文へ戻るようになっています。このようなプログラムは無限ループと呼ばれ終りがありません。プ

ログラムを止めるには BREAK キーを使うしかありません。ただ、20行のINPUT 文で、変数に数値が入力されるのを待っている状態がありますから、プログラムの流れは中断することになります。

コーヒーブレイク

プログラムの１行とは？

プログラムを画面に書いていきますと、１つの行番号で画面の１行におさまらないで、５～６行といった長いプログラムになるときもあります。このような長いプログラムでも、プログラムは１行といいます。

しかし、同じ「行」では紛らわしいので、画面の１行を「物理行」、１つの行番号に続くプログラムを「論理行」と呼ぶこともあります。

そして、この「論理行」の長さは行番号も含めて256字以内でなければなりません。257字以上書いた場合は、257字以降の文が無効になります。

4．走り出したプログラムを止めるには

走っている（実行している、またはRUNの状態にある）プログラムを途中で止めるには、BREAK キーを使うことは前に述べました。

しかし、BREAKはプログラムの中では使うことはできません。プログラムを実行させ、あるところでプログラムを止めるための命令をプログラムの中に書き込むことができます。それがSTOP命令です。

STOP、CONT

リストの35行は30行までの計算を終了したところでプログラムを止めるために30行と40行の間に書かれたものです。

```
10 A=5:B=7:C=3
20 X=A*2
30 Y=B-1
35 STOP
40 Z=X+Y
50 PRINT Z
```

```
RUN
Break in 35
CONT
 16
Ready
```

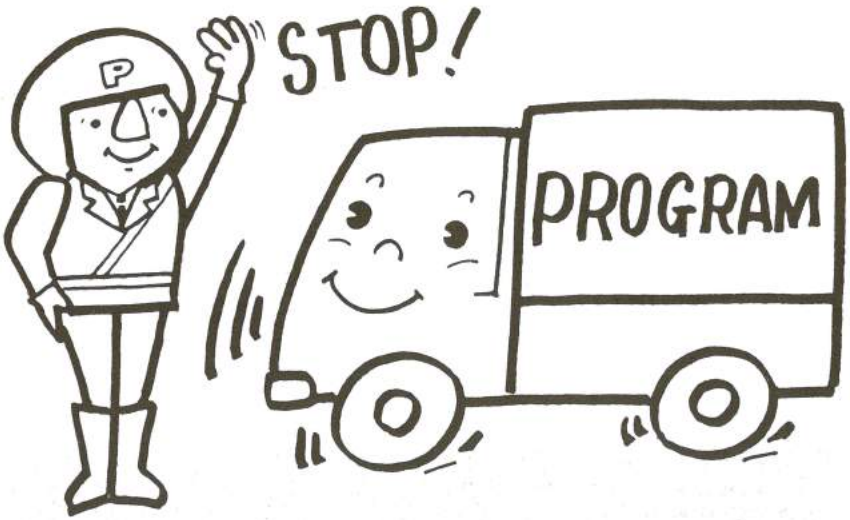
このプログラムをRUNさせますと、Break in 35と表示され、35行でプログラムが中断されていることがわかります。そのあとCONTでプログラムの実行が再開され、40行の計算が行なわれ、50行でPRINTされました。答の16が表示されています。

どうしてこのようにプログラムを途中で止める必要があるのかといいますと、上の例では、Zの値がどうもおかしい、自分の考えている値とコンピュータの出した答が一致しないという場合に、一つ前の段階のXとYの値を調べてみようというときに、XとYの計算が終わった30行のあとでコンピュータを一たん止めてみるのがよいからです。

X、Yの値を調べるのは簡単ですね。前にお話ししましたダイレクト・モードを使えばよいのです。

?X CR

?Y CR ですね。



第4章　コンピュータらしい仕事をさせるプログラム

この章あたりから、「コンピュータを使っている」という感じがしてきます。いままではどちらかというと、タイプライターや電卓とたいして変らなかったと思います。

なぜコンピュータらしくなるかというと、同じことを何回も繰り返すことをやらせるとか、計算だけでなく、判断をさせるからです。

BASICはむずかしい、とよく言われますが、BASICを初めて勉強しようとする人にとってもっともむずかしい命令は、この章で扱うFOR～NEXTとIF～THENではないかと思います。これら2組の命令が上に述べた繰り返しと判断のための命令で、プログラムの中にはひんぱんに使われますので、ぜひともマスターしなければなりません。

しかし、BASICを学ぶには、全部の命令を覚える必要はありません。もっとも基本的な命令を10～15くらいマスターし、あとは必要になったときに本で調べる程度でよいのです。ただ、どのような働きをする命令があるのかということぐらいは、あらかじめ知っておいて欲しいと思います。

1．同じ仕事を繰り返すFOR～NEXT

一定回数、繰り返して仕事をさせるときに使います。

さっそくプログラムを入力してみましょう。FOR～NEXT文の含まれるもっとも簡単なプログラムです。実行結果からわかるように、1から5までの数値を表示するプログラムです。

```
10 FOR I=1 TO 5
20 PRINT I
30 NEXT I
40 END
```

```
RUN
 1
 2
 3
 4
 5
Ready
```

FOR～NEXT 文は一般に次の形で表わされます。

```
FOR  I= 1   TO   5   STEP   1
  ↗   ↑         ↑          ↑
変数名  初期値   最終値      増分   (増分が１の場合は省略可)
```

代入文

NEXT I

変数名は何でもよいのですか、I,J などがよく使われます。

初期値(しょきち)から最終値(さいしゅうち)まで繰り返すのですが、初期値、最終値は変数や数式でもかまいません。

STEP は、初期値から最終値までいくつおきに増えていくか（または減っていくか）を決めるものです。STEP～は省略できます。省略した場合は増分は１になります。

NEXT Iの Iは省略できます。

では、少し変えたプログラムで実験してみましょう。

```
10 A=10
20 FOR I=A TO A+16 STEP 3
30 PRINT I;
40 NEXT
50 END
```

```
RUN
 10  13  16  19  22  25

Ready
```

FOR I=A TO A+16 STEP 3 と変わりました。10行でA=10となっていますから、FOR I=10 TO 26 STEP 3と同じことです。

PRINT Iの次に；(セミコロン)をつけましたので、実行結果は画面のとおり、横に数値が並びました。

FOR文の中をもう少し変えてみましょう。

初期値より最終値の方が大きい方が自然ですが、プログラムによっては逆に最終値の方が小さいこともあります。その場合はSTEPの次の増分はマイナスの数値になります。なお、初期値や増分は整数でなくてもよく、小数点がついてもかまいません。

```
10 A=12.5
20 FOR I=A TO -A STEP -2.5
30 PRINT I;" ";
40 NEXT
50 END
```

```
RUN
 12.5  10  7.5  5  2.5  0 -2.5 -5 -7.5
 -10 -12.5
```

もう説明はいらないと思います。30行の""(ダブル・クォーテーション)は、RUNさせた結果の表示が見やすいように、数値間のスペースをとるためです。

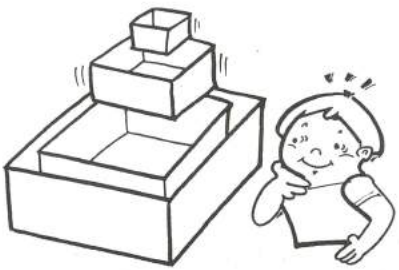
FOR～NEXT 文は次のように、重ねることができます。

```
10 FOR J=1 TO 5
20 FOR I=1 TO 3
30 PRINT "A";
40 NEXT I:PRINT
50 NEXT J
60 END
```

このプログラムを実行しますと次のようになります。

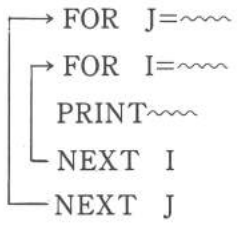
```
RUN
AAA
AAA
AAA
AAA
AAA
Ready
```

I=1 TO 3で"A"を3つ並べます。3つ並べ終ったところで改行し(40行のNEXT Iの次のPRINT 文)、それを5回繰り返します。

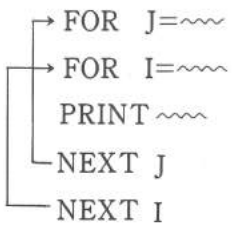


FOR～NEXT 文を重ねる場合、「入れ子にする」といいます。上のイラストのように、大きい箱の中に少し小さい箱が入り、その箱の中にまたやや小さい箱が入るようになっています。FOR～NEXT 文も何重にも重ねる場合に、きちんと内側に、そのまた内側に入っていくような形にしないといけません。

〈正しい使い方〉



〈正しくない使い方〉



矢印の線が交差するのはいけません。

また、矢印の中に外から割って入ってくるのもいけません。しかし、矢印の中から外へ抜け出すのはかまいません。

FOR～NEXTの〝入れ子〟は8重まで有効です。

次に、FOR～NEXTを使ったプログラムをご紹介しましょう。

FOR～NEXT文を使ったプログラム〈九九表〉

```
10 FOR J=1 TO 9
20 FOR I=1 TO 9
30 IF LEN(STR$(I*J))=2 THEN A$=" "+STR
$(I*J):GOTO 50
40 A$=STR$(I*J)
50 PRINT A$;
60 NEXT I:PRINT
70 NEXT J
80 END
```

```
RUN
  1  2  3  4  5  6  7  8  9
  2  4  6  8 10 12 14 16 18
  3  6  9 12 15 18 21 24 27
  4  8 12 16 20 24 28 32 36
  5 10 15 20 25 30 35 40 45
  6 12 18 24 30 36 42 48 54
  7 14 21 28 35 42 49 56 63
  8 16 24 32 40 48 56 64 72
  9 18 27 36 45 54 63 72 81
Ready
```

30行はケタを揃えるためのプログラムです。この行に使われている命令は後章にゆずります。

2．合格、不合格を振り分けられるIF～THEN

コンピュータは計算をするだけでなく、「判断」もします。ある数が100より大きいか、30より小さいか、または負の数か、といったこと、さらに数ばかりでなく、ある文字（列）が〝ABC〟という文字列と同じかというようなことを判断します。

IF～THEN……は英語で「もし～ならば……」という意味です。コンピュータでは、IF～THENによる仕事のことを条件判断といいます。

次のプログラムはテストか何かの点数を入力しますと、コンピュータが自動的に65点以上なら合格、65点未満から不合格と判断するプログラムです。

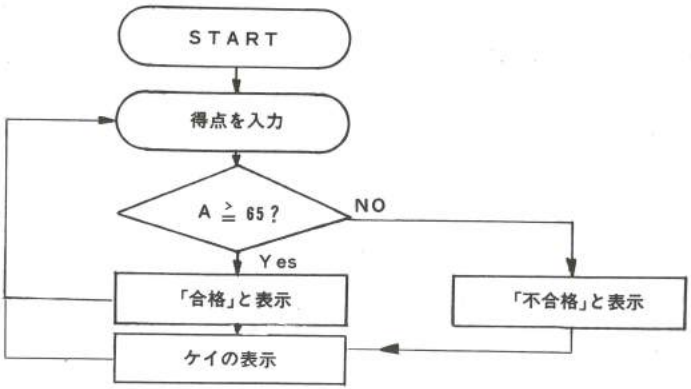
```
10 INPUT "ﾄｸﾃﾝ=";A
20 IF A>=65 THEN 40
30 IF A<65 THEN 50
40 PRINT "ｺﾞｳｶｸ":GOTO 60
50 PRINT "ﾌｺﾞｳｶｸ"
60 PRINT "----------"
70 GOTO 10
```

```
RUN
ﾄｸﾃﾝ=86
ｺﾞｳｶｸ
----------
ﾄｸﾃﾝ=64.9
ﾌｺﾞｳｶｸ
----------
ﾄｸﾃﾝ=
Break in 10
```

上のプログラムはちょっとわかりにくいと思いますので、日常私たちが使っていることばで書き直してみましょう。

```
10  得点を入力して下さい
20  入力した得点が65点以上ならば40行へ
30  入力した得点が65点未満ならば50行へ
40  「合格」と表示、そして60行へ
50  「不合格」と表示
60  「――――――――」を表示
70  10行へ
```

これをフローチャート（流れ図）に描きますと、次のようになります。



IF～THEN 文にはいくつかのルールがあります。これらのルールをしっかり覚えれば、もう BASIC はこわくありません。あとはスイスイと進むと思います。

上のプログラムの20行に IF～THEN　40　とあるように、THEN のあとに行番号がくることが多いのです。行番号のほかに、PRINT～とか A＝B＊C などの代入文がきてもよいのです。実は、THEN のあとにすぐ行番号がくるのは、GOTO が省略されているのです。

IF 文は一般的に次のように説明されます。

```
I F  A>=65     THEN  GOTO  40
     論理式            文
```

そして、論理式が成り立つ（上の場合はAが65以上）場合はTHEN以降を実行し、逆に論理式が成り立たない（Aが65未満）場合はTHEN以降の文を無視して、次の行の実行に移ります。

さて、もう少し前のページのプログラムを見て下さい。40行にGOTO 60とありますが、これを書き忘れるとどうなるでしょう。

これを忘れますと、得点が65以上のとき、ゴウカクと表示され、続けてフゴウカクと表示されてしまいます。ゴウカクと表示したらフゴウカクは飛び越えなければなりませんから、GOTO 60が必要です。

IF～THENの後にくる文の例を挙げておきます。

IF～THEN　PRINT〝XXX〟
IF～THEN　END
IF～THEN　STOP
IF～THEN　BEEP　（ブザーを鳴らします）
IF～THEN　GOSUB　行番号　（サブルーチンへ飛びます）

サブルーチンについては後で述べます。

3．IF～THENを使って入力ミスを防ぐプログラムを作る

入力ミスをなくすように作られているプログラムはよいプログラムです。この章に出てきた合格、不合格を判断するプログラムで、入力する数値が0点から100点までの間に限られているのなら、この範囲外の数値を入力したときはもう一度入力をやり直すことができるようにすれば、よいプログラムになります。

次ページのプログラムの15行は、上の目的のために追加した行です。得点が100を超えている、またはマイナスの数値（負数）の場合には10行に戻って、もう一度得点の入力待ちになります。

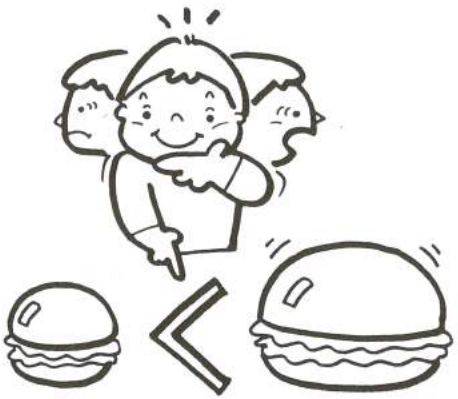
A＞100　OR　A＜0　の書き方は「Aが100より大きいか、またはAが0より小さいか」ということですから、BASICは本当に人間の言葉に近づいた、わかりやすい言語といえましょう。

このプログラムに出てくるORについては、次の論理演算子を参照して下さい。

	記号	使い方	意味
比較演算子	=	A＝B	AとBは等しい
	<>	A＜＞B	AとBは等しくない
	＞	A＞B	AはBよりも大きい
	＜	A＜B	AはBよりも小さい
	＞＝	A＞＝B	AはBに等しいか，Bより大きい
	＜＝	A＜＝B	AはBに等しいか，Bより小さい
論理演算子	NOT	NOT A＞8	Aが8より大きくない
	AND	A＝3 AND B＞5	Aが3で，同時にBが5より大きい
	OR	A＞100 OR A＜0	Aは100より大きいか，または0より小さい
	XOR	A＞10 XOR B＞10	AかBのどちらか一方だけが10より大きい

IF～THEN文に登場する論理式に使われる＜とか＞＝の記号は比較演算子といいます。

比較演算子の中の＝（等号、イコール）は、LET（代入文）の中で使われる＝と異なり、ふだん私たちが使っている＝です。同じ記号でもコンピュータは使われる場所によって区別します。頭がいいですね。



```
10 INPUT "トクテン=";A
15 IF A>=100 OR A<0 THEN 10
20 IF A>=65 THEN 40
30 IF A<65 THEN 50
```

```
40 PRINT "ｺﾞｳｶｸ":GOTO 60
50 PRINT "ﾌｺﾞｳｶｸ"
60 PRINT "----------"
70 GOTO 10
```

```
RUN
ﾄｸﾃﾝ=108
ﾄｸﾃﾝ=-12.6
ﾄｸﾃﾝ=66
ｺﾞｳｶｸ
----------
ﾄｸﾃﾝ=
Break in 10
```

4．プログラムを短縮させるGOSUBとRETURN

次のプログラムを見て下さい。四則演算のプログラムですが、それぞれの計算の下に点線（マイナス記号の連続）を入れるようにしてあります。30、50、70、90の各行は110行と同じPRINT文を書いてもよいのですが、同じ文を何回も書くことを避けるために、GOSUBとRETURNをペアで使います。

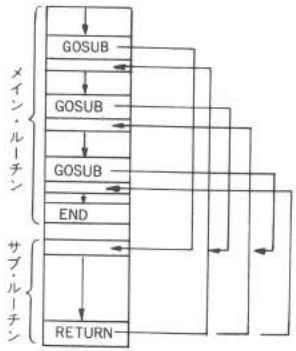
```
10 INPUT "A,B ﾃﾞ ﾆｭｳﾘｮｸ ｼﾅｻｲ ";A,B
20 PRINT "ﾀｼｻﾞﾝ...";A+B
30 GOSUB 110
40 PRINT "ﾋｷｻﾞﾝ...";A-B
50 GOSUB 110
60 PRINT "ｶｹｻﾞﾝ...";A*B
70 GOSUB 110
80 PRINT "ﾜﾘｻﾞﾝ...";A/B
90 GOSUB 110
100 END
110 PRINT "----------"
120 RETURN
```

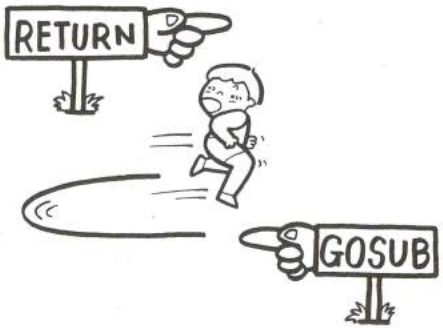
```
RUN
A,B ﾃﾞ ﾆｭｳﾘｮｸ ｼﾅｻｲ 5
ﾀｼｻﾞﾝ... 17
----------
```

```
ヒキザン....7
カクザン.... 60
ワリザン.... .41666666667
```

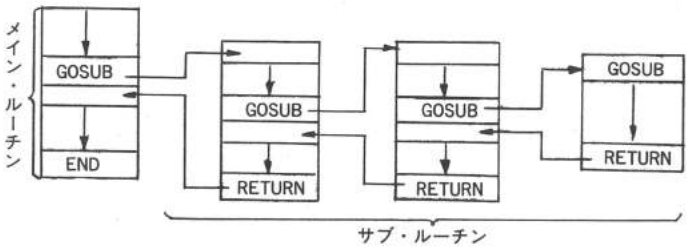
さて、GOSUB 文によって指定された行番号以降の文（上のプログラムは110行と120行）のことをサブ・ルーチンといいますが、このサブ・ルーチンをどこに置いたらよいかが問題になります。別にどこに置いてもよいのですが、普通はプログラムの最後の方に置きます。100行の END 文のあとなら、もっともわかりやすいのではないのでしょうか。

そして、サブ・ルーチンの最後は RETURN で締めくくります。次の図のように、サブ・ルーチンの仕事が終わったあとは、メイン・ルーチン（サブに対して、本流のプログラム）の GOSUB 文の次の行番号へ戻ります。





GOSUB 文は次の図のように、サブ・ルーチンの中でも使えます。この場合、16重まで有効です。



5．メニューから品物を選ぶときは ON～GOTO、ON～GOSUB

IF 文と GOTO 文とをいっしょにしたような、ON～GOTO 文というのがあります。

喫茶店でコーヒーとかケーキとかを品物の名前でなく、メニューに書いてある品物の番号で注文するといったことに似ています。ウェイトレスが1番ならコーヒー、2番ならケーキと判断するのですが、コンピュータなら、これを ON～GOTO 文が判断するのです。

次のプログラムはウェイトレスの「ご注文は？」の問に対して、メニューの番号で欲しい品物を答えると、ウェイトレスが「○○○ですね」と確認してくれるものです。「ゴチュウモン　ハ？」に対して3を入力すると「ミルクデスネ」と映ります。

15行は、1～3以外の数字を入力した場合に、もう一度入力をしなおす指示を出すためです。


```
10 INPUT"ｺﾞﾁｭｳﾓﾝ ﾊ ?";A
15 IF A<1 OR A>3 THEN PRINT "1~3 ﾃﾞﾓｳｲ
ﾁﾄﾞ ";:GOTO 10
20 ON A GOTO 100,200,300
100 PRINT "ｺｰﾋｰ ﾃﾞｽﾈ":GOTO 10
200 PRINT "ｹｰｷ ﾃﾞｽﾈ":GOTO 10
300 PRINT "ﾐﾙｸ ﾃﾞｽﾈ":GOTO 10
```

```
RUN
ｺﾞﾁｭｳﾓﾝ ﾊ ?5
1~3 ﾃﾞﾓｳｲﾁﾄﾞ ｺﾞﾁｭｳﾓﾝ ﾊ ?2
ｹｰｷ ﾃﾞｽﾈ
ｺﾞﾁｭｳﾓﾝ ﾊ ?1
ｺｰﾋｰ ﾃﾞｽﾈ
ｺﾞﾁｭｳﾓﾝ ﾊ ?
Break in 10
```

なお、ON～GOTO 文は IF 文と GOTO 文とをいっしょにしたような、と書きましたが、実際に、ON～GOTO を使わないで、IF 文と GOTO 文を使って、上のプログラムを書き替えることができます。これはみなさんでやってみて下さい。

ON～GOTO と同じような働きをする命令に ON～GOSUB があります。GOSUB を使ったときは、サブ・ルーチンの最後に RETURN と書くのを忘れないように。

ON～GOSUB を使って、前のプログラムを書き直すと、次のようになります。実行結果は同じになります。

```
10 INPUT"ｺﾞﾁｭｳﾓﾝ ﾊ ?";A
15 IF A<1 OR A>3 THEN PRINT "1~3 ﾃﾞﾓｳｲ
ﾁﾄﾞ ";:GOTO 10
20 ON A GOSUB 100,200,300
30 GOTO 10
100 PRINT "ｺｰﾋｰ ﾃﾞｽﾈ":RETURN
200 PRINT "ｹｰｷ ﾃﾞｽﾈ":RETURN
300 PRINT "ﾐﾙｸ ﾃﾞｽﾈ":RETURN
```

```
RUN
ｺﾞﾁｭｳﾓﾝ ﾊ ?9
1~3 ﾃﾞﾓｳｲﾁﾄﾞ ｺﾞﾁｭｳﾓﾝ ﾊ ?2
ｹｰｷ ﾃﾞｽﾈ
ｺﾞﾁｭｳﾓﾝ ﾊ ?3
ﾐﾙｸ ﾃﾞｽﾈ
ｺﾞﾁｭｳﾓﾝ ﾊ ?
Break in 10
```

第5章　データ処理入門

前章ではコンピュータにコンピュータらしい仕事をさせる命令について学びましたが、こんどは大量のデータを扱うということで、なお一層コンピュータらしい仕事をさせることを覚えましょう。

1．プログラム上のデータを読み込む命令

READ, DATA（リード、データ）

READ、DATA はプログラムの中で常にペアで使われます。ですからREAD、DATA の使われている命令文のことを「READ～DATA 文」と呼ばれます。

四則演算をする場合のように、その都度数値を入力する場合には READ～DATA 文は不要ですが、常に同じ数値を使って仕事をさせる場合とかプログラムとデータを切り離したくないときなどは、この READ～DATA 文が必要になってきます。

```
10 READ A,B,C,D
20 PRINT A+B+C+D
100 DATA 1,2,3,4
```

```
RUN
 10
```

上のプログラムでは100行の DATA 文に 1、2、3、4 という数値データが入っています。これを10行の READ 文の中の変数で読み出しています。したがって、上のプログラム・リストの10行と100行は、次のプログラム10行と同じことになります。

10 A＝1：B＝2：C＝3：D＝4

なお、DATA 文は、READ 文の後にあっても先にあってもよいのです。次の例は読みとるデータが文字の場合です。

```
10 READ A$,B$,C$,D$
20 PRINT A$+B$+" "+C$+" "+D$
30 DATA SEGA , PERSONAL
40 DATA COMPUTER
50 DATA SC-3000
```

```
RUN
SEGA PERSONAL COMPUTER SC-3000
```

上の例では、DATA 文に数字を入れても文字として扱われますから、計算には使えません。

READ～DATA 文で扱う DATA の数と READ 文の変数の数はふつうは一致させますが、一致していないときはどうなるでしょう。

READ 文の変数の数＞DATA 文のデータの数……エラー

READ 文の変数の数＜DATA 文のデータの数……変数の数しか読まない

READ 文は、DATA 文の最初から順に読んでいきますから、READ 文の変数の数が少ないときは、DATA 文の最後の方を読み残してしまいます。

RESTORE（リストア）

RESTOREは同じデータを始めから繰り返して読む場合に使います。

40行のREAD　Eは初めのデータ、1を読みます。

```
10 READ A,B,C,D
20 DATA 1,2,3,4
30 RESTORE
40 READ E
50 PRINT A+B+C+D+E
60 END
```

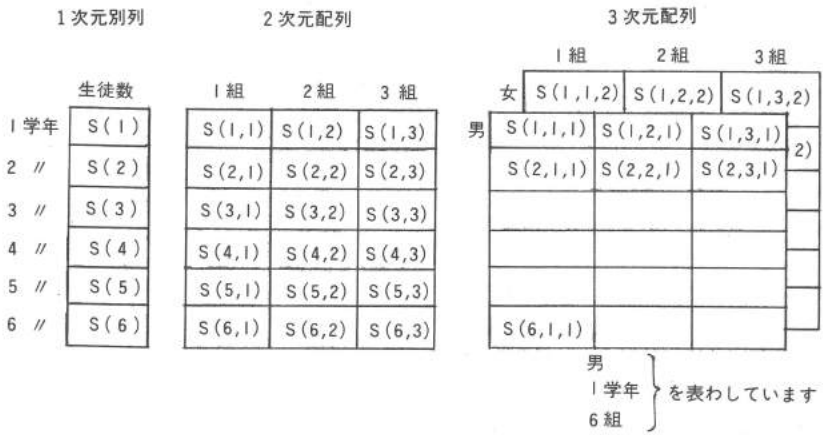
```
RUN
 11
```

2．数値が多いときは「配列」を使う

DIM

配列というのは数多くの数値または文字列を規則正しく整理してしまっておくメモリーの集まりです。

配列には「1次元配列」「2次元配列」「3次元配列」の3つがあります。次の例を見て、1次元～3次元の違いを理解して下さい。



1次元配列

READ, DATA文のところのプログラムでは4つのデータに対してA, B, C, Dと変数を使いました。しかし、データが増えるごとに変数を一つずつ設定するのはたいへんです。

こんなとき、配列を使うと便利です。配列は配列宣言文DIM～を使って設けます。配列を設けることを「配列を宣言する」と言います。

DIM　B(5) カッコの中の数値を添字といいます。

この意味は、B(0)、B(1)、B(2)、B(3)、B(4)、B(5)の6個の変数を設けたことになります。

配列には文字変数も使えます。文字変数の配列を宣言するには、

DIM　A$ (5)

というように書きます。

```
10 CLS
20 DIM A$(5),B(5)
30 FOR I=0 TO 5
40 READ A$(I),B(I)
50 PRINT A$(I),B(I)
60 PRINT
70 NEXT I
80 END
100 DATA コーヒー,300,ミルク,150,ケーキ,200
110 DATA ティー,280,トースト,180,パン,100
```

```
RUN
コーヒー          300

ミルク            150

ケーキ            200

ティー            280

トースト          180

パン              100
```

DIM A$(5)としたのに、READ文でA$(I)になっているのは、FOR文の中で使っているIです。ですからIが0から5に変化しながら、データを読み込んでいきます。

READ文では、A$とBをペアで読み込みますから、DATA文の中でも文字列と数値を交互に並べます。

2次元配列

2次元配列はDIM(5, 5)のように、カッコの中の添字が2つになります。

かけ算の九九表は2次元配列の見本のようなものです。

次のプログラム・リストはかけ算の九九表の答の数値を配列に一度全部を

```
10 CLS
20 DIM A(9,9)
30 FOR J=1 TO 9
40 FOR K=1 TO 9
50 A(J,K)=J*K
60 NEXT K,J
70 FOR J=1 TO 9
80 FOR K=1 TO 9
90 PRINT A(J,K);
100 NEXT K
110 PRINT
120 NEXT J
130 END
```

```
 1  2  3  4  5  6  7  8  9
 2  4  6  8  10  12  14  16  18
 3  6  9  12  15  18  21  24  27
 4  8  12  16  20  24  28  32  36
 5  10  15  20  25  30  35  40  45
 6  12  18  24  30  36  42  48  54
 7  14  21  28  35  42  49  56  63
 8  16  24  32  40  48  56  64  72
 9  18  27  36  45  54  63  72  81
Ready
```

しまって、その後配列からひとつずつ拾い出して画面に表示するようにしています。

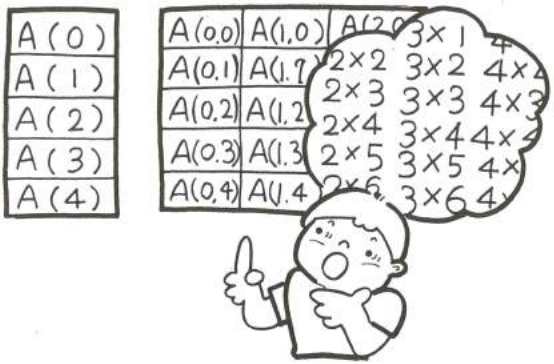
(47ページの九九表は、配列にしまうということをしないで、計算するとすぐに画面に表示するようにしています。)

このプログラムでは、桁を揃えるための行がありません。桁を揃えるには、47ページのプログラムを参考にして下さい。

3次元配列

3次元配列はDIM(5, 4, 8)のように、添字が3つになります。添字の数字はメモリーの許す限り、いくつでもかまいませんが、3次元になりますと、例えば全部の添字が10の場合には、10×10×10で1,000のメモリーが必要です。また、添字が全部20の場合は20×20×20で8,000メモリーが必要です。数値の場合、1メモリーに10バイト必要ですから8,000メモリーは80,000バイト必要ということになり、メモリー容量を完全にオーバーしてしまいますから、配列の大きさには注意しましょう。

なお、配列は3次元までしかできません。また、配列宣言をしないで添字のある変数を使った場合は、添字の数が10以下なら有効です。このことを「暗黙の配列宣言」といいます。



ERASE(イレーズ)

プログラムの途中で配列宣言を無効にしたい時に使います。なぜ途中で無効にする必要があるのかといいますと、コンピュータのメモリは有限である

ために途中で不要になった配列は削除して、新しく必要になった配列を追加して、メモリーを有効に使うためです。

ERASE としますと、ERASEより前にあった配列は全部無効になります。

ERASE W のように配列名を書きますと、その配列を無効にします。

```
10 DIM W(11)
20 FOR I=1 TO 11
30 READ W(I)
40 PRINT W(I);
50 NEXT I:PRINT
60 ERASE
65 FOR I=1 TO 5
70 PRINT W(I);
75 NEXT
80 DATA 11,10,9,8,7,6,5,4,3,2,1
```

```
RUN
 11  10  9  8  7  6  5  4  3  2  1
 0  0  0  0  0
```

上のプログラム・リストでは60行にERASEが書かれていますので、W（11）の配列は消されました。次の70行で再びW（I）の配列を使いますが、60行のERASEによって配列がなくなっていますから、PRINTしても0しか出ません。

第6章　プログラミングを便利にする命令

1．不要な行を消す

DELETE（デリート）

プログラムの不要な行を消す方法として、30 CR とすると行番号30のプログラムが消せることは前に述べました。しかし、たくさんの行をいっぺんに消したい場合には、DELETE 命令を使うと便利です。

- ○ DELETE　80－130　CR　　行番号80から130までのプログラムを消去
- ○ DELETE　－240　CR　　先頭から行番号240までのプログラムを消去
- ○ DELETE　1000－　CR　　行番号1000以降のプログラムを消去
- ○ DELETE　500　CR　　行番号500を消去

－のかわりに、, (コンマ)を使ってもかまいません。つまり、DELETE 30

—70と DELETE30, 70は同じです。

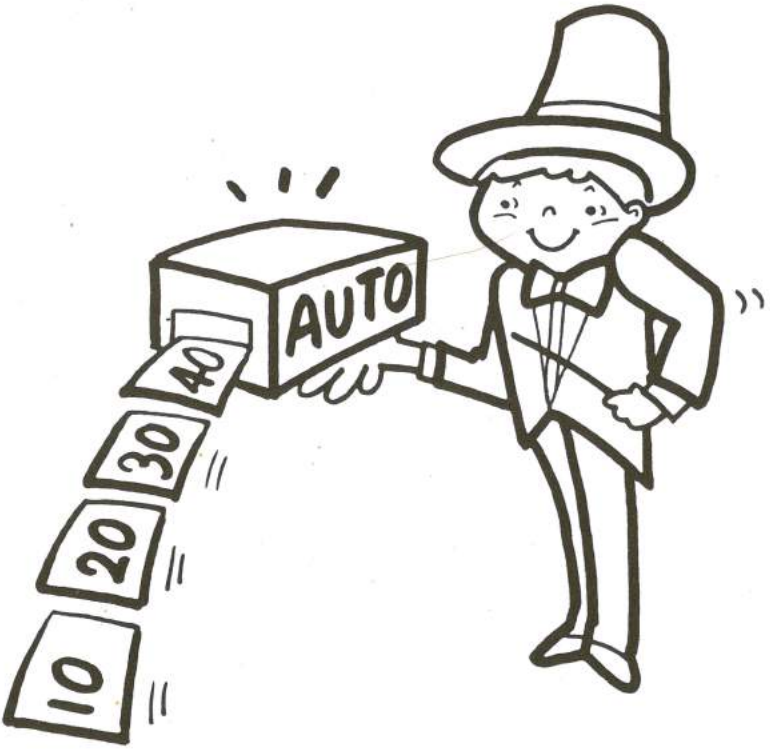
1行だけ消去する時は、DELETEを使わなくてもできます。つまり、DELETE 60 CR と、60 CR とは同様です。

2．行番号を自動発生させる

AUTO（オート）

BASICでプログラムを作る場合、必ず行番号を入れなければなりませんでしたね。SC—3000シリーズでは、行が変わるごとに、行番号を打ち込む手間をはぶくため、行番号を自動発生させることができます。

AUTO CR と入力して下さい。次の行に10と表示されましたね。ここに行番号10のプログラムを入力し、CRを押すと、次に20と表示されます。



このようにして、行番号10から10行きざみで自動的に行番号が発生してきます。この自動発生を止めたいときは、BREAKキーを押して下さい。

また、次のように命令すると、発生する行番号を好きなように指定できます。

AUTO 1000 CR　行番号1000から、10行きざみに発生

AUTO 100, 20 CR　行番号100から、20行きざみに発生

発生開始行番号　きざみ幅

3．行番号を付け直す

RENUM（リナンバー）

プログラミングをしているとき、行と行の間に追加をしているうち、詰まり過ぎてもう追加できなくなってしまうことがときどき起こります。そのときには、RENUMで行番号を付け直し、行番号のきざみ幅を大きくすることにしましょう。

RENUM CR　プログラムの先頭から、10、20、30、……と行番号が付け直される

RENUM 500 CR　プログラムの先頭から、500、510、520、……と行番号が付け直される

RENUM 1000, 300 CR　行番号300からのプログラムを、行番号1000から10きざみに付け直される。

RENUM 1000, 300, 20 CR　行番号300からのプログラムを、行番号1000から20きざみに付け直される。

プログラムを作り終え、行番号を整頓するときにもRENUMは使われます。

4．プログラム・リストを見やすくする

REM（レム）

プログラム中に注釈文を入れておくと、後からプログラム・リストを見たときわかり易いです。注釈文として例えば、プログラムの題名、ブロックごとのプログラムの内容、サブ・ルーチンの内容、プログラム製作者名、製作年月日など書いておくと便利です。注釈文はREMの後に書きます。

```
10 REM 2スウ ノ タシザン
20 INPUT A
30 INPUT B
40 PRINT A+B
50 GOTO 20
```

REM文で書かれたことは、プログラムの実行中、無視されます。

第7章　カセット・テープにプログラムやデータを保存する

1．プログラムの保存

SAVE（セーブ）、VERIFY（ベリファイ）

キーボードからプログラムを入力するという作業は、意外と大変なことですが、一所懸命苦労して入力したプログラムも、本体の電源を切ってしまうと、きれいさっぱり消滅してしまいます。そこで、必要なプログラムは、なんらかの形で保存することにしましょう。保存方法はいろいろありますが、ここではデータ・レコーダーを本体に接続して、カセット・テープにプログラムを保存する方法を紹介します。

まず、

SAVE "ファイル・ネーム"　CR

と入力して下さい。ファイル・ネームはプログラムの名前のことです。" 以外の文字であれば英字でもカタカナでも何でも構いませんが、16字以内でなければなりません。また、スペースも1文字と数えられ、区別されます。例えば、〝GAME　1〞と〝GAME 1〞というファイル・ネームは、コンピュータは違うものとして解釈します。ファイル・ネームは後から見て便利なように、プログラムの内容が判るような名前を付けておいた方がよいでしょう。さて、次の行に、

＊ Saving start

と表示されましたね。そうしたら、データ・レコーダの録音ボタンを押して下さい。これで、いま、コンピュータが記憶しているプログラムがカセット・テープに保存されていくのです。「SAVE」は「カセット・テープにプログラ

ムを保存せよ」という命令で、保存することをセーブするといいます。しばらくして、画面に

＊ Saving end

と表われたら、セーブ終了です。カセットを止めて下さい。

セーブが終ったからといって安心して、これで本体の電源を切ってしまってはいけません。セーブをした後には、本当に正しくセーブされたかどうか、必ずチェックしましょう。カセット・テープをセーブを始めたところまで巻き戻し、

VERIFY [CR]

と入力して下さい。そして、カセット・デッキのプレイ・ボタンを押すと、カセット・テープからセーブしたプログラムを見つけて、コンピュータの記憶しているプログラムとのチェックが行なわれます。正確にセーブされていた場合は、画面に＊ Verifying end と表示されますが、

? Verifying error

などと表示され、最終的に＊ Verifying end と出てこなかった場合は、きちんとセーブされなかったということです。このときには、きちんと接続されているかどうかなど確認し、もう一度、セーブからやり直して下さい。

プログラムをセーブしたカセット・テープには、ファイル・ネームと、データ・レコーダのカウンタ・ナンバーの何番から何番まで入っているか書いておきましょう。そうしないと、カセット・テープのどこに何のプログラムが入っているかわからなくなってしまって、またそのカセット・テープを使う時、とても不便ですから。

2．プログラムの再生

LOAD（ロード）

今度は、セーブしたプログラムをカセット・テープから再生してみましょう。まず、カセット・テープをそのプログラムがセーブされているところまで巻き戻し、

LOAD "再生したいプログラムのファイル・ネーム" CR

と入力し、次の行に、

＊ Loading start

と現われたら、カセット・デッキのプレイ・ボタンを押して下さい。画面に、

Found "ファイル・ネーム"

と表示されたら、コンピュータがカセット・テープからそのプログラムを見つけ、再生を開始したことになります。しばらくして、

＊Loading end

と表示されたら、再生終了ですから、カセットを止めて下さい。もし、

？ Memory writing error

などと表示され、うまくできなかったときには、接続を確認したり、データ・レコーダの音量を調整して、やり直しましょう。

また、カセット・テープにたくさんのプログラムをセーブしていて、どこに再生したいプログラムがあるかわからなくなっても、テープを最初まで巻き戻して、同様にやってみて下さい。すると、関係のないプログラムはどんどん読みとばしていき、指定したプログラムが再生されます。

Skip "ファイル・ネーム"

は、「プログラムを見つけたけれども、読みとばしますよ」という意味です。

「LOAD」は、「カセット・テープからプログラムを呼び出す」という命令です。プログラムを呼び出すことをロードするといいます。

ロードするときに、ファイル・ネームを付けずに

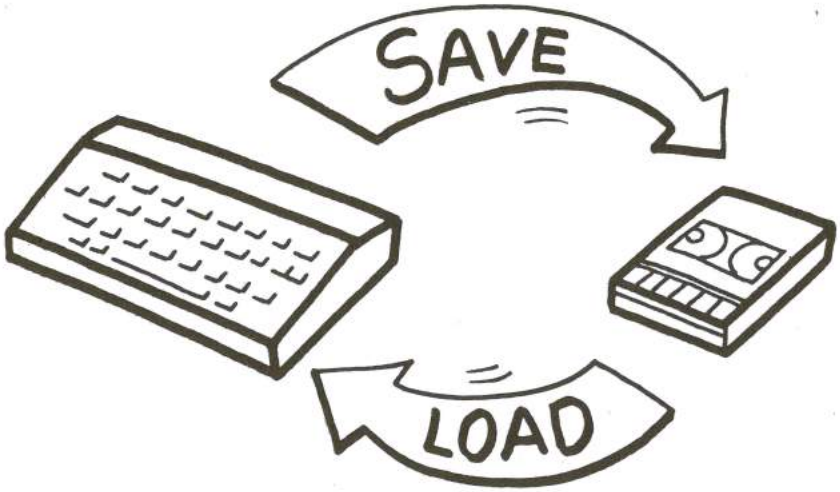
LOAD CR

とすると、カセット・テープで最初に見つけたプログラムがロードされます。ですから、カセット・テープに１つしかプログラムがセーブされていない場合、またはカセット・テープの最初からセーブされているプログラムを順にロードしていきたい場合は、ファイル・ネームを付けなくても構いません。

※カセット・デッキは、SC—3000専用のもの以外に、お手持ちのカセット・テープレコーダーも使えます。そのときには、トーンや音量調整をして、

うまくセーブ・ロードできる音量レベルを見つけて下さい。(但し、お手持ちのカセット・デッキで、使用できないものもあります。)

※カセット・テープは、コンピュータ専用のもの以外に、音楽等を録音したりするのに使うカセット・テープでも使えます。



第8章　コンピュータのおもしろさが倍増する「関数」

1．ゲームに欠かせない乱数（らんすう）とインテジャー

RND（ランダム）

SC—3000シリーズには、乱数を発生させる機能があります。この乱数を使えば、サイコロの目を出したり、ゲームで標的を不規則に動かしたりなどいろいろなおもしろいことができます。乱数を発生させる関数は、RNDですが、引き数は1、0、－1の3通りです。

RND（1）　0以上1未満の乱数が続けて発生（毎回違う数字が発生）

RND（0）　乱数系列が初期化（毎回同じ数字が発生）

RND（－1）　乱数系列を入れ変える（新しい乱数のグループを作る）

では、10個の乱数を出すプログラムを作ってみましょう。

```
10 FOR I=1 TO 10
20 R=RND(1)
30 PRINT R
40 NEXT I
```

このプログラムを RUN してみましょう。

```
10 FOR I=1 TO 10
20 R=RND(1)
30 PRINT R;
40 NEXT I
RUN
 .41977382483  .089899732351  .952260398
04  .44826552472  .71496988052  .59593800
008  .80387178485  .1244595121  .23572802
072  .77037796814

Ready
```

0以上1未満のでたらめな数が10個出てきましたね。このでたらめな数のことを乱数といいます。

INT（n）（インテジャー）

乱数を使えば、コンピュータをサイコロがわりに使うことができます。サイコロは、1～6の目がでたらめに出るものです。ですから、コンピュータに、1～6の整数をでたらめに発生させればよいわけです。

ところで、RND（1）は、0以上1未満の数をでたらめに発生させるものでした。つまり、

$$0 \leqq \mathrm{RND}\ (1) < 1$$

ということです。この RND(1) を使って、1～6の整数を発生させることを考えてみましょう。そこで使われるのが、INT 関数です。INT (n) は、nの小数点以下を切り捨てるという関数です。ですから、RND（1）を6倍したもの、つまり、

RND（1）＊6

INT 関数の中に入れ、INT (RND(1)＊6) とします。これで0～5の整数の乱数を発生するわけです。しかし、サイコロの目は1～6の整数でした。ですから、

INT (RND (1)＊6)＋1

とすればいいわけですね。では、これを使って、サイコロの目を次々出すプ

ログラムを作ることにしましょう。

```
10 S=INT(RND(1)*6)+1
20 PRINT S;
30 GOTO 10
```

```
RUN
 5 1 4 6 3 1 6 2 5 5 6 6 2 1 1 5 5 5 4
 2 6 4 6
Break in 30
```

こうして考えていくと、a～bまでの整数の乱数を発生させる式は、

INT (RND (1)*(b−a))+a

となりますね。a、bにいろいろな数値を入れて試してみて下さい。

2．文字列に使われる関数

ASC（〝n〞）（アスキー関数）

SC—3000シリーズで使うことのできるすべての文字や記号について、それぞれ番号がついています。たとえば、〝A〞が65、〝a〞が97、〝ア〞が177となっていて、これらの番号を「キャラクターコード」と呼びます（P153「キャラクターコード」表参照）。

ASC(〝n〞)は、その文字を対応するキャラクターコードに変換する関数です。ダイレクト・モードで

? ASC (〝A〞) CR

と実行すると、次の行にAのキャラクターコードである65が表示されましたね。ASC(関数を使うときには、("A")のように、必ずカッコ内の文字を〝〞で囲んで下さい。また、ASC(K$)のように、文字列変数を使っても、もちろん構いません。もしも、K$が2文字以上の文字列になっていても、最初の

1文字のキャラクターコードがASC(K$)の値となります。ですからK$＝〝SEGA〟の場合、AS$（K$）は〝S〟のキャラクターコード、83です。

コンピュータは、人間のように文字や記号をそのまま判断することはできません。すべての文字や記号は、コンピュータ内部で数字に変換されて扱っているのです。ですから私達も、コンピュータで文字や記号を使うとき、キャラクターコードで使うこともできます。

CHR$（キャラクターダラー）

ASC関数と反対の働きをするのがCHR$関数です。CHR$(n)で、数値nに対応する文字や記号、コントロール機能が与えられます。例えば、ダイレクト・モードで

? CHR$(65) CR

```
LIST

10 FOR M=32 TO 255
20 PRINT CHR$(M);
30 NEXT M

RUN
 !"#$%&'()*+,-./0123456789:;<=>?@ABCDE
FGHIJKLMNOPQRSTUVWXYZ[¥]^ｧ`abcdefghijk
lmnopqrstuvwxyz{|}~ ｢｣ｰｱｲｳｴｵｶｷｸｹｺｻ
ﾀﾁﾂﾃﾄﾅﾆﾇﾈﾉﾊﾋﾌﾍﾎﾏﾐﾑﾒﾓﾔﾕﾖﾗﾘﾙﾚﾛﾜﾝﾞﾟ
Ready
```

と命令すると、キャラクターコードが65である A が表示されますね。では、次のプログラムを実行して、コンピュータの中に入っている文字や記号をのぞいてみましょう。

キーボードに印字されている文字や記号が並んで出て来ましたね。P154の「キャラクターセット」と比較すると同じになっています。これが、SC-3000シリーズで表示できる文字・記号のすべてです。これ以外の文字・記号を表示したい場合は、自分で作っていかなければなりません（P96PATTERN参照）。

LEFT$（レフトダラー）、RIGHT$（ライトダラー）、MID$（ミドルダラー）

次の3種類の関数を使うことによって、長い文字列の中から、文字の一部を取り出すことができます。

LEFT$ (A$, n)　文字列 A$の左から n 番目までの文字を取り出す。

RIGHT$ (A$, n)　文字列 A$の右から n 番目までの文字を取り出す。

MID$ (A$, m, n)　文字列 A$の左から m 番目の文字を起点として n 文字取り出す。

```
10 A$="SEGA SC-3000"
20 PRINT"A$・・・・・";A$
30 PRINT"LEFT$(A$,4)・・・・・・";LEFT$(A$,4
)
40 PRINT"RIGHT$(A$,4)・・・・・";RIGHT$(A$,
4)
50 PRINT"MID$(A$,6,2)・・・・・";MID$(A$,6,
2)
```

これらを使ったプログラムの実行結果は次のようになります。

```
RUN
A$・・・・・SEGA SC-3000
LEFT$(A$,4)・・・・・・SEGA
RIGHT$(A$,4)・・・・・3000
MID$(A$,6,2)・・・・・SC
Ready
```

スペースも１文字として数えられることに注意して下さい。

また、文字列は、わざわざ文字変数に置き換えず、LEFT$("SEGA　SC-3000", 4）としても結果は同じです。

LEN（レングス）

LEN(A$)で、A$の文字数を数えることができます。この場合も、〝で囲まれた文字数全部で、スペースも含まれます。ですから、もし、〝の間が全部スペースであっても文字として扱われ、スペースの個数が数えられるわけです。

実際にLENを使ったプログラムを実行してみましょう。

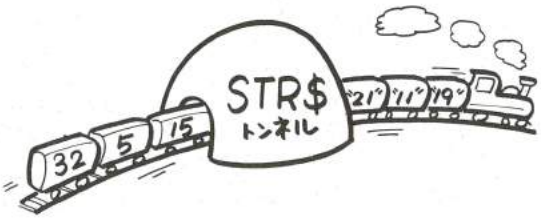
```
LIST

10 INPUT"A$=";A$
20 PRINT"LEN(A$)=";LEN(A$)
30 GOTO 10
```

```
RUN
A$=SEGA SC-3000
LEN(A$)= 12
A$=ｺﾝﾋﾟｭｰﾀ
LEN(A$)= 7
A$=
```

スペースも含んで文字の数を数えてくれてますね。

STR$（ストリングダラー）

STR$で、数値を文字列に変換することができます。



```
10 INPUT"A=";A
20 INPUT"B=";B
30 PRINT"A+B=";A+B
40 D$=STR$(A)+STR$(B)
50 PRINT"STR$(A)+STR$(B)=";D$
```

```
RUN
A=12
B=34
A+B= 46
STR$(A)+STR$(B)= 12 34
Ready
```

STR$ (A)、STR$ (B) とすると、変数 A、B に入っている数値12、34が、それぞれ文字列 "12"、"34" となります。ですから上のプログラムの行番号40では、"12"＋"34" という文字列のたし算が行なわれ、文字がくっつくだけで 1234 となるのです。34の前のスペースは＋の符号が省略されたためです。

VAL（バリュー）

VAL は、STR$とまったく正反対の働きをする関数で、数字の文字列を数値に変換します。ですから、VAL("123")は123という数値になります。VAL と STR$はペアで覚えておきましょう。

3．時刻を表示

TIME$（タイムダラー）

SC—3000シリーズには、水晶発振（すいしょうはっしん）の正確なクォーツ・デジタル時計の機能が内臓されています。本体の電源を ON にしたときを「00：00：00」として、その時から１秒きざみで働き始めます。では、いま、電源を入れてからどれぐらい時間が経ったかを見てみることにしましょう。その時間を知ることができる関数が TIME$です。

? TIME$ CR

とダイレクト命令をすると時間が表示されますね。例えば、

01：54：38

と表示された場合、電源をONにしてから1時間54分38秒経ったことを示しています。

この時計機能を使えば、SC—3000をデジタル時計がわりに利用することができます。そのプログラムは、次のようです。

```
10 INPUT"ﾌﾞｻﾞｰ ｾｯﾄ TIME ﾊ ? ";B$
20 INPUT"ｲﾏ ﾉ TIME ﾊ ? ";T$
30 TIME$=T$:CLS
40 CURSOR 12,15:PRINT TIME$
50 IF TIME$<>B$ THEN 40
60 CURSOR 10,15:PRINT"ｼﾞｶﾝ ﾃﾞｽﾖ !!"
70 FOR I=0 TO 5:BEEP 2:NEXT I
80 CLS:GOTO 40
```

まず、ブザーを鳴らしたい時刻と今の時間をセットします。その時刻が8時3分15秒であれば、

08：03：15 CR

という形で入力して下さい。行番号30では、TIME$関数に現在の時刻をセットしています。このコンピュータ・デジタル時計は、本体の電源を切るか、RESETボタンを押すまで働きます。

また、TIME$は、時間を競うゲームなどにも使うことができます。ゲームスタート時に、

TIME$= "00：00：00"

とプログラム中で初期状態にセットし直して使えばいいわけです。

4．スペースを作る関数

SPC（スペース）

いままでスペース（空白）をプリントするとき、PRINT文の " " の中で、スペースの個数分スペース・バーを押してプリントしていました。2つか3つのスペースの場合はこの方法でもよいのですが、これが10や20のスペースをプリントするということになれば、ちょっと大変ですね。そういうときには、SPC関数を使うことにしましょう。SPC（n）は、n個のスペースの文字列を与えてくれる関数です。次のプログラムのように、SPC関数はPRINT文の中で使われます。

```
10 PRINT"ABC";SPC(10);"XYZ"
```

```
RUN
ABC          XYZ
Ready.
```

実行すると、CとXの間に10文字のスペースがあきます。またSPCで指定した範囲に何か文字が書かれていると、消されてしまいます。

TAB（タビュレーション）

TAB関数は、画面の左端から何文字目に出すか指定する関数です。これも、PRINT文で使います。

次のプログラムでは、左5文字文のスペースをあけ、ABCと表示します。

```
10 PRINT TAB(5);"ABC"
```

```
RUN
     ABC
Ready
```

TAB関数の場合、SPC関数と違って、指定した範囲に文字があっても消しません。SPCのときは、実際にスペースをプリントしているのに対して、TABの場合は、画面左端から次に表示するところまで移動するだけで、スペースはプリントされていません。

このTAB関数は、表を書くときなどによく利用されます。その一例を載せておきましょう。

```
10 PRINT TAB(5);"ｶﾓｸ";TAB(11);"ﾃﾝｽｳ"
20 PRINT TAB(4);"--------------"
30 PRINT TAB(5);"ｺｸｺﾞ";TAB(12);70
40 PRINT TAB(5);"ｻﾝｽｳ";TAB(12);82
50 PRINT TAB(5);"ﾘｶ";TAB(12);91
60 PRINT TAB(5);"ｼｬｶｲ";TAB(12);64
```

```
RUN
     ｶﾓｸ     ﾃﾝｽｳ
    --------------
     ｺｸｺﾞ      70
     ｻﾝｽｳ      82
     ﾘｶ        91
     ｼｬｶｲ      64
Ready
```

5．どのキーが押されたか判断する関数

INKEY$（インキーダラー）

INKEY$は、文字や数字のどのキーが押されたかを見る関数です。このINKEY$関数は、ゲームのとき、キーボードの指示によって何かを動かしたりなどに使うと便利です。

次ページのプログラムは、スペースキーを押すと☺が、その他のキーを押すと ○ が画面に表示されるというプログラムです。

行番号30の" "は、ヌルストリング（空の文字列）です。この場合K$がヌ

ルストリングのときは、何も表示せず、行番号10へ戻り、キーの入力待ちをします。

```
10 K$=INKEY$
20 IF K$=" " THEN PRINT "●";:GOTO40
30 IF K$<>"" THEN PRINT "○";
40 GOTO 10

RUN
```

6．残りのバイト数がわかる

FRE（フリー）

SC—3000シリーズ本体の電源をONにしたとき、またはRESETボタンを押したとき、画面に、

26620 Bytes free

と表示されますね。このとき、SC—3000シリーズは1つもプログラムやデータを記憶していない状態で、メモリー（プログラムやデータを記憶することができる量）が、26620バイトあることを示しています。しかし、プログラム

やデータを入力していくと、このバイト数はどんどん減っていきます。あと、どれぐらいメモリーが残っているかを知るには、FRE関数で調べることができます。ダイレクト命令で

? FRE CR

と命令したとき、

8300

と表示されたなら、残りのバイト数が8300で、あと8300バイトのプログラムやデータを入れることができることを示しています。

第9章　絵や図表を描くための命令

1．画面を選択する

SCREEN（スクリーン）

これまで何気なく、画面にプログラムを書いたり、文字を表示させたりしてきましたが、実はSC—3000シリーズには2種類の画面があります。1つはテキスト画面、もう1つはグラフィック画面で、これまで使っていた画面は、テキスト画面の方です。プログラムや文字を書くテキスト画面に対し、グラフィック画面は、文字の他、細かい模様を描くことができます。私たちは、この2つの画面に自由に書き込んだり、表示したりできます。書き込む画面と実際に画面に表示している画面は必ずしも同じでなくても構いません。ですから、テキスト画面を表示しているときに、グラフィック画面に模様を描いていくこともできるわけです。どちらの画面を選択するかは、

SCREEN　書き込み画面, 表示画面

（書き込み画面、表示画面は、1または2の番号で指定

1：テキスト画面　　2：グラフィック画面）

で指定します。しかし、これまでのようにテキスト画面しか使わない場合には、SCREEN指定は必要ありません。

2．表示位置を決める

CURSOR（カーソル）

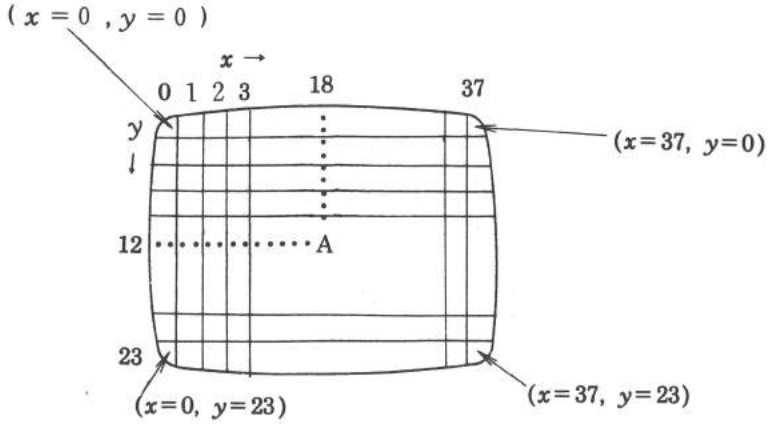
プログラムを実行すると、これまでは表示が左側に出るだけでした。これを、見やすくするために、画面上のどの位置に表示するか、プログラム上で

指定することができます。その表示位置を決める命令が、

CURSOR　X座標，Y座標

です。画面上はいくつかのマス目に分れていて，そのマス目に文字を表示したり、ぬりつぶしていって絵を描いたりすているわけです。(X座標，Y座標,)は、マス目の位置を表わす座標です。画面の座標構成は、テキスト画面、グラフィック画面によって異なります。

○テキスト画面座標



テキスト画面は、38桁×24行の912のマス目で構成されています。

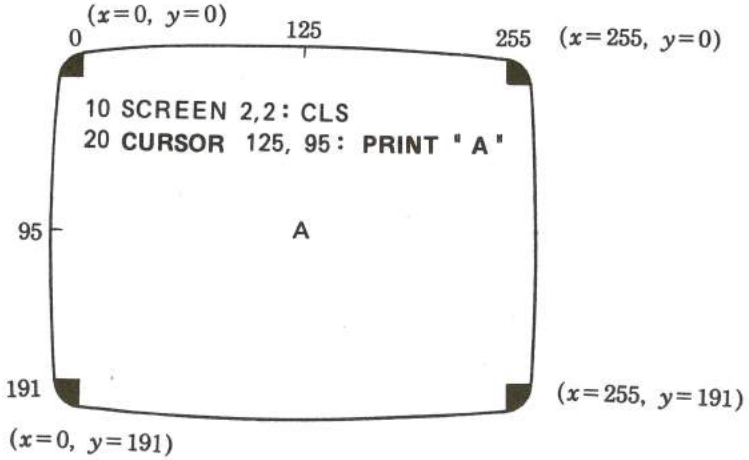
CURSOR 18, 12：PRINT〝A〞 CR

とダイレクト命令をすると、Aの文字が画面の中央に表示されましたね。この位置が、テキスト画面での座標（18，12）です。

では、このCURSORと、前章でやったINKEY$を使って、ひとつおもしろいプログラムを作ってみましょう。←キーを押すとHは1マス左へ、→キーを押すと右へ動くプログラムです。そうするためには、つまり、←キーが押されていれば、HのX座標を1減らし、→キーが押されていれば、1増やせばいいわけです。行番号70のPRINT文の中で、" H "とすることに注意して下さい。スペースをプリントすることによって、これで移動前のHが消されるのです。

```
10 CLS:X=18:Y=20
20 K$=INKEY$
30 IF K$=CHR$(28) THEN X=X+1
40 IF K$=CHR$(29) THEN X=X-1
50 IF X<0 THEN X=0
60 IF X>33 THEN X=33
70 CURSOR X,Y:PRINT" H "
80 GOTO 20
```

○グラフィック画面座標



x 軸方向　0～255　(256ドット)
y 軸方向　0～191　(192ドット)

グラフィック画面は、細かい点（ドット）によって構成されています。グラフィック画面で、CURSOR文で座標を指定すると、表示したい文字の先頭位置がきまります。

3．色をつける

COLOR（カラー）

カラーテレビ、カラーディスプレイを使用している場合は、文字やキャラクター、下地の色を、SC—3000で使うことができる16色の中から自由に指定

できます。色はCOLOR命令で、それぞれ色に対応する0～15の色番号で指定します。

色と色番号の対応

色番号	色	色番号	色	色番号	色
0	透　明	6	こい赤	12	こい緑
1	黒	7	水色(シアン)	13	マゼンダ
2	緑	8	赤	14	灰
3	うすい緑	9	うすい赤	15	白
4	こい青	10	こい黄		
5	うすい青	11	うすい黄		

○テキスト画面の場合

COLOR　m　,　n　　(m、n：0～15)

m：文字の色

n：下地の色

BLINE、PSETにより線や点、塗りつぶされた部分が消された時の色。

○グラフィック画面の場合

COLOR　a　,　b　,　(X1,　Y1)－(X2,　Y2),　c

a、b、c：0～15　X1、X2は0～255　　Y1、Y2は0～191

a：PRINT文による文字の色

LINE、PSET文による線や点の色

LINE、PAINT文による塗りつぶしの色

b：背景色

BLINE、PRESETにより線や点、塗りつぶされた部分が消された時の色

c：バックドロップカラー

(X1,　Y1)と(X2,　Y2)を結ぶ直線を対角線とする長方形の範囲を塗りつぶす。

背景色の下にあるのがバックドロップ・カラーです。電源をONにした直

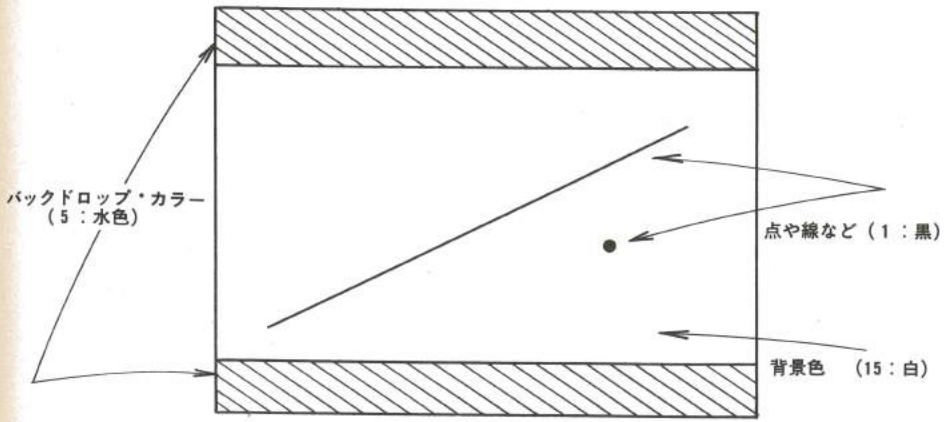
後の、グラフィック画面のカラー指定の状態は、

COLOR 1, 15, (0, 0) — (255, 191), 5

で、下図のようになっています。画面上下の色と、点や線などを描ける中央の部分の色とは違っていますが、これと同じにするためには、

COLOR 1, 15, (0, 0)— (255, 191), 15

のように、バックカラーとグランドカラーの色を同じにして指定すればよいのです。いまの例では、下地が白、点や線などは黒となります。



4．線を引く

LINE（ライン）

グラフィック画面に線を引いてみましょう。線を引くためには、書き始めの座標（X1，Y1）、書き終わりの座標（X2，Y2）とその線の色Cを指定しなければなりません。

LINE (X1, Y1)— (X2, Y2), C

X1，X2：0～255

Y1，Y2：0～191

C：0～15

LINE文で連続した線を描く場合には、書き始めの座標（X1，Y1）は

省略できます。また、線の色も省略でき、そのときはCOLOR文で指定した色で線が引かれます。

では、次のプログラムをRUNしてみましょう。

```
10 SCREEN 2,2:CLS
20 X1=20:Y1=1:X2=240:Y2=190
30 LINE(X1,Y1)-(X2,Y1),I
40 FOR I=1 TO 14
50 LINE-(X2,Y1),I
60 LINE-(X2,Y2),I
70 LINE-(X1,Y2),I
80 LINE-(X1,Y1+6),I
90 X1=X1+8:Y1=Y1+7:X2=X2-8:Y2=Y2-7
100 NEXT I
```

RUNして、グラフィック画面に線を描終わると、すぐにテキスト画面に変わってしまいましたね。グラフィック画面を見たいときには、SHIFTキーを押しながらCTRLキーを押して下さい。画面がグラフィック画面に変わり、線が描かれているのがわかりますね。

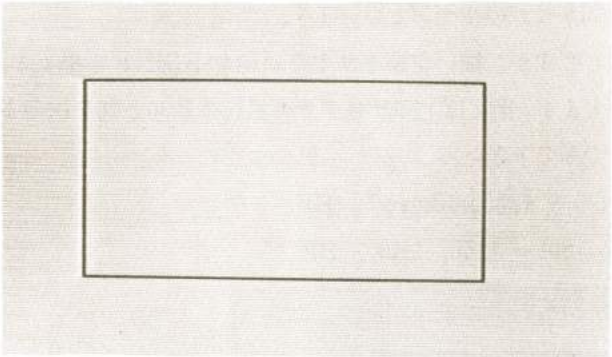
ではもう1度、SHIFTキーを押しながらCTRLキーを押してみましょう。画面がテキスト画面に戻りましたね。

では次に、LINE文で箱(BOX)を描いてみましょう。箱を描くには、LINE文で対角線を指定して、色の後にBを付けます。

LINE (X 1, Y 1)− (X 2, Y 2), C, B

これで、座標(X 1, Y 1)と(X 2, Y 2)を結ぶ直線と対角線とする箱が描けます。では、(80, 50)と(160, 100)を結ぶ直線を対角線とする箱を、実際に描いてみましょう。

```
10 SCREEN 2,2:CLS
20 FOR C=0 TO 15
30 LINE(60,50)-(200,130),C,B
40 FOR I=0 TO 300:NEXT I
50 NEXT C:GOTO 20
```



箱の色が次々に変化していきますね。このプログラムは、行番号50で無限ループにしてあります。ですから、止めるときには BREAK キーを押して下さい。

さて、今度は、いまのプログラムの行番号30のBをBFにして、RUNしてみて下さい。すると、箱の中が塗りつぶされましたね。

LINE (X 1, Y 1)− (X 2, Y 2), C, BF

は、(X 1, Y 1)と(X 2, Y 2)を結ぶ直線を対角線とする箱の中をCで指定した色で塗りつぶす命令です。

BLINE（ビーライン）

BLINEで、LINE文で描いた線や箱を消すことができます。使い方は、LINE文と同じですが、色指定は無視されます。

次のプログラムは、斜めに線を引き、数秒後にその線を消し、また違う色で

線を引くというのをくりかえすプログラムです。

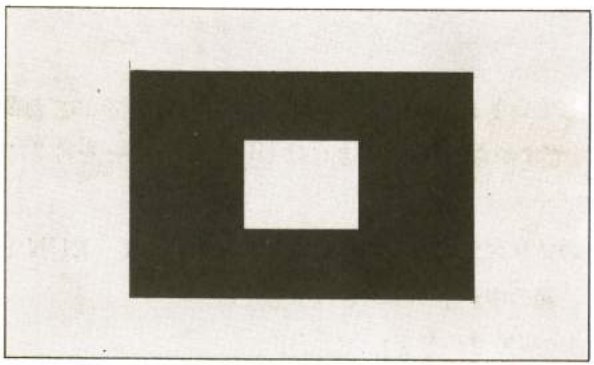
```
10 SCREEN 2,2:CLS
20 FOR C=0 TO 15
30 LINE(60,50)-(200,130),C
40 FOR I=0 TO 300:NEXT I
50 BLINE(60,50)-(200,130)
60 FOR J=0 TO 300:NEXT J
70 NEXT C:GOTO 20
```

このプログラムの行番号40と60のように FOR と NEXT の間で何もしていないループを、時間かせぎのループといいます。

箱を消す場合も同じですが、描いた箱より小さい箱を BLINE で描くと、その部分だけ色が消えます。例えば上のプログラムの行番号30と50を次のように変えて、実行してみて下さい。

30 LINE (50, 50)－ (200, 150), 1, BF

50 BLINE (100, 80)－ (150, 120),, BF



行番号50で[,][,]となっているのは、[,]と[,]の間で指定する色指定を省略しているのです。

BLINE 文は、一度描いた絵を全部消したり、一部分だけ消すのに使います。

5．円を描く

CIRCLE（サークル）

CIRCLE 命令で、とてもきれいな円を描くことができます。CIRCLE 文の使い方はやや複雑ですが、慣れるとそうでもありません。

CIRCLE（X，Y），半径，色，比率，始点，終点（，BまたはBF)

①中心座標（X，Y）　X：0～255　Y：0～191

②半径　中心点からの半径

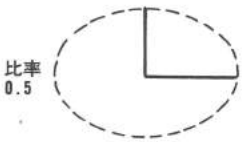
③色　0～15まで

④比率　縦と横の半径の比率

1のとき　真円

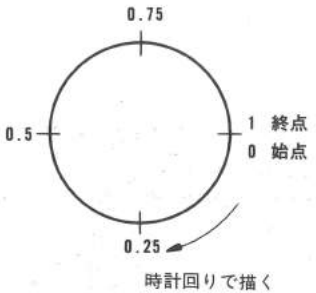
1より小さい　横長の円

1より大きい　縦長の円



⑤始点　書き始めの位置　0～1

⑥終点　書き終りの位置　0～1



⑦BまたはBF

この部分で何も指定しないときは円周のみ描く

BFにすると③で指定した色で塗りつぶす

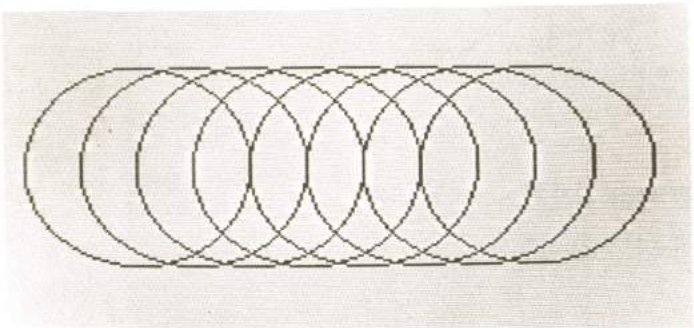
Bを指定すると内側にも線が引かれる

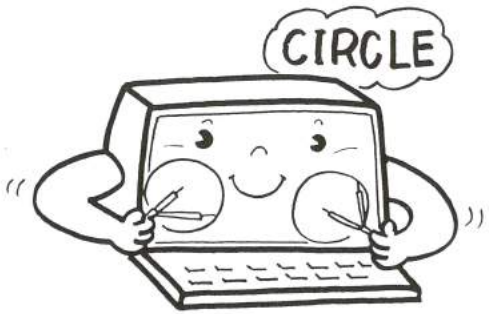
例）始点0.25、終点0.5の場合

図－６

次のプログラムを実行すると、円を横にずらして描いていきます。

```
10 SCREEN 2,2:CLS
20 FOR X=60 TO 200 STEP 20
30 CIRCLE(X,95),40,8,1,0,1
40 FOR I=0 TO 300:NEXT I
50 NEXT X
```





BCIRCLE（ビーサークル）

BCIRCLEは、CIRCLE文で描いた円を消すときなどに使います。指定方法は同じですが、BCIRCLEの場合、色指定は無視され、下地と同じ色で円を描くので見えなくなります。

次のプログラム は前のプログラムに60行以下を追加したものです。

```
10 SCREEN 2,2:CLS
20 FOR X=60 TO 200 STEP 20
30 CIRCLE(X,95),40,8,1,0,1
40 FOR I=0 TO 300:NEXT I
50 NEXT X
60 FOR X=60 TO 200 STEP 20
70 BCIRCLE(X,95),40,,1,0,1
80 FOR I=0 TO 300:NEXT I
90 NEXT X
```

一度描かれた円が、左側から１つずつ消えていきますね。

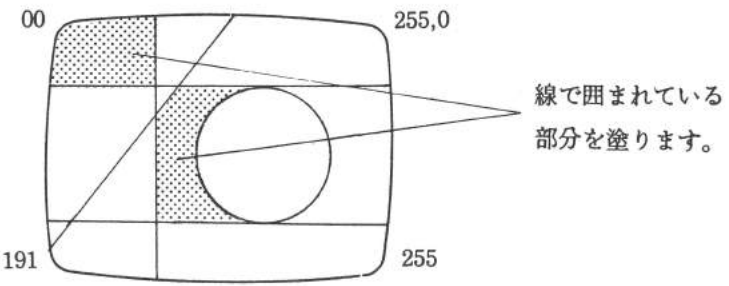
6．区切られた部分を塗りつぶす

PAINT（ペイント）

画面に、LINE文やCIRCLE文で区切られた部分を、好きな色で塗りつぶすことができます。その命令は、

PAINT（x, y), 色番号

で、座標（x, y）から塗り始め、（x, y）を含む囲まれている部分を指定した色で塗りつぶします。



線で区切って、そこを塗りつぶす場合、すき間を作らないように注意して下さい。

7．画面に点を打ったり消したりする

PSET（ピーセット）

グラフィック画面に、1つのドットを表示してみることにしましょう。ドットを表示する命令は、

PSET（X，Y），色番号

です。

次のプログラムをRUNすると、グラフィック画面の中央の座標(127，95)に、1つのドットが黄色で表示されましたね。

```
10 SCREEN 2,2:CLS
20 PSET(127,95),10
30 GOTO 30
```

PSETを使って、座標を連続的に変化させると直線や曲線(きょくせん)が描けます。

また、PSETは、点をちりばめたり、三角関数などの関数を使ったグラフの作成などにも使うことができます。

PRESET（ピーリセット）

PRESETは、PSETと反対に点で消す命令です。

PRESET (X,Y)

という形で使い、座標 (X, Y) を点で消します。

前のプログラムに、次の2行を加えて実行してみて下さい。

```
22 FOR I=0 TO 50:NEXT I
24 PRESET(127X95)
```

黄色い点が映った思ったら、すぐに消えましたね。

8．座標の原点とプラスの方向が決めれる

POSITION（ポジション）

グラフィック画面には、画面の左上を0として、右へX座標が0～255まで、下へY座標が0～191まで、というように座標が決められていましたが、POSITIONで座標を指定することができます。

POSITION (X,Y)，Xの軸方向，Yの軸方向

(x,y) の座標が新しい座標の中心点、x＝0、y＝0となる。

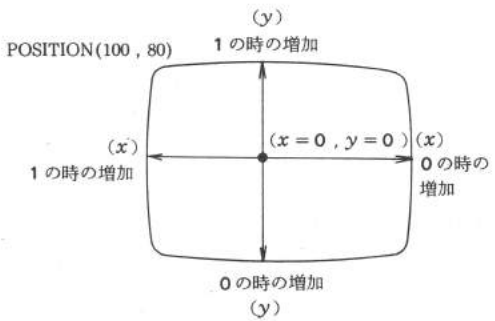
x軸の方向、y軸の方向は0または1で指定

0：xは右に増加

　yは下に増加

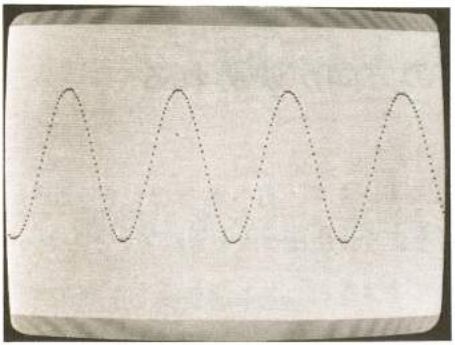
1：xは左に増加

　yは上に増加



POSITION 文は PSET と一緒に使うと関数のグラフなどが描けます。次のプログラムは、それらを使ってサイン・カーブを描くプログラムです。

```
10 SCREEN 2,2:CLS
20 POSITION(100,50),0,0
30 FOR N=-20 TO 5 STEP .1
40 X=N*10+100:Y=SIN(N)*50+50
50 PSET(X,Y),1
60 NEXT N
```



9．文字や絵のパターンが作れる

PATTERN（パターン）

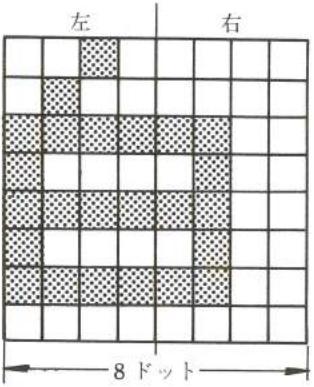
PATTERN 文を使うと、新しく文字や絵（キャラクター）を作り出すことができます。

PATTERN　C#　文字コード，〝文字列式〟

文字コード：32～255または&H20～&HFF

文字列式：16進数で指定

では、¥を押すと、テキスト画面に「白」が表示されるというプログラムを作ってみましょう。そのためには、白という漢字を16進数に直していかなければなりません。



左	右	左	右
0 0 1 0	0 0 0 0	2	0
0 1 0 0	0 0 0 0	4	0
1 1 1 1	1 1 0 0	F	C
1 0 0 0	0 1 0 0	8	4
1 1 1 1	1 1 0 0	F	C
1 0 0 0	0 1 0 0	8	4
1 1 1 1	1 1 0 0	F	C
0 0 0 0	0 0 0 0	0	0

黒いマス＝1，白いマス＝0

¥の文字コードは&H5C（16進数で5Cという意味。P ⑦「キャラクタセット参照」）です。そこで次のようなプログラムになります。

```
10 PATTERN C#&H5C,"2040FC84FC84FC00"
```

これをRUNして、¥キーを押してみて下さい。白が表示されましたね。それは、¥に白が書き込まれたからです。同じような方法で、漢字などの新しい文字や記号を作り出すことができます。

では、もう一度、使い方を整理しておきましょう。

○ C#　テキストモード指定

○文字コード　16進数の場合、&H20～&HFFまでの数値で指定

　　10進数の場合、32～255までの数値で指定

○文字列式　8×8のマス目ドットを黒く塗りつぶして、文字や図形を書く。

黒い点は1、白い点は0として、各列毎に1と0を割り当てる。

たとえば は、01110000となる。

次に上4桁、下4桁に分け、16進数に変換。

例の場合、0111は7、0000は0で"70"となる。こうして変換した数を、上の列から順につなげていく。

グラフィック画面に出す絵も、同じようにしてできます。

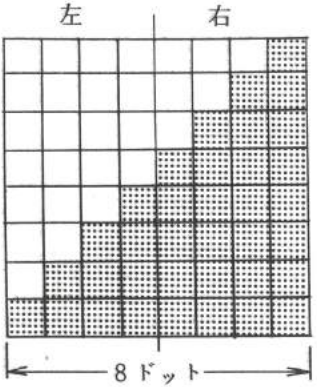
PATTERN　S#　スプライト名称,〝文字列式〟

○　S#　　グラフィック画面指定

○　スプライト名称　　0～255までの番号が付けられる

○　文字列式　　テキスト・モードの場合と同様

```
10 SCREEN 2,2:CLS
20 PATTERN S#0,"0103070F1F3F7FFF"
30 SPRITE 0,(10,0),0,1
```



0000	0001	0	1
0000	0011	0	3
0000	0111	0	7
0000	1111	0	F
0001	1111	1	F
0011	1111	3	F
0111	1111	7	F
1111	1111	F	F

このプログラムをRUNすると、グラフィック画面に◢が表示されましたね。行番号30のSPRITE命令については、P100で説明しますが、PATTERN命令とペアで使う命令です。

〈パターンの描き方〉

自分でパターンを作っていくとき、8×8のマス目に仕切られた用紙を使うと便利です。そのマス目を塗りつぶし、パターンを作っていきます。

パターンができたら、塗った所を1、空白を0として、マス目の横に1列ずつ、0や1の数字を並べます。

その数字を上4桁、下4桁に区切り、それぞれ16進数に直します。直すときは、次の10進数、2進数、16進数の対応表を参考にして下さい。

各列ごと直された2桁の16進数を、上の列からつなげて、文字列として〟

の中に代入します。

文字は8×6、グラフィックは8×8で表現されます。

10進数	2進数	16進数
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F
16	10000	10

10. スプライト機能を使う

MAG（マグ）

PATTERN文でスプライト面に描かせる絵の大きさは4種類あり、

MAG n　　n:0～3

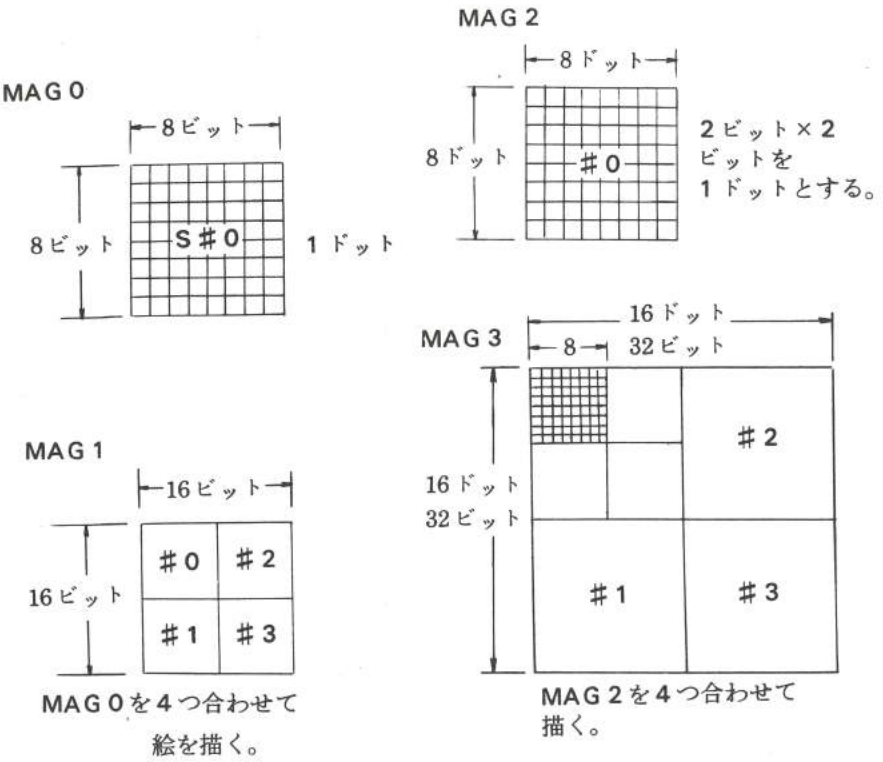
で指定します。

MAG 0：1ビットを1ドットとして、8×8ドットの枠内に絵を描く

MAG 1：8×8ドットのパターンの4個組(#0～#3、#4～#7、……, #252～#255) を結合して、16×16ドットの枠内に絵を描く。

MAG 2：2×2ビットを1ドットとして、8×8ドットの枠内に絵を描く。つまり、ビット数は16×16ビットになる。

MAG 3：MAG 2で指定した枠を4つ合わせて描くことができ、32×32ビットになる。



SPRITE（スプライト）

SC−3000には、シリーズには、スプライト機能があります。スプライト機能とは、グラフィック画面上でパターンを自由に動かすことができる機能です。この機能を使うためには、前に説明したMAG文、PATTERN文、それからいまから説明するSPRITE文は絶対に欠かせないものです。

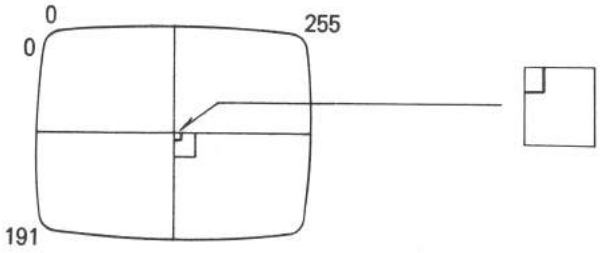
SPRITE　面番号，(X，Y)，スプライト名称，色番号

○面番号　0～31

スプライト面は、前から0番、1番、……、31番の32枚の面があるので、どの面に描くかを番号で指定。スプライト同志が交差するときは、番号が小さい方が前になる。

○（X，Y）

グラフィック画面の座標を使い、基準はMAG命令で指定した枠の一番左上の座標。



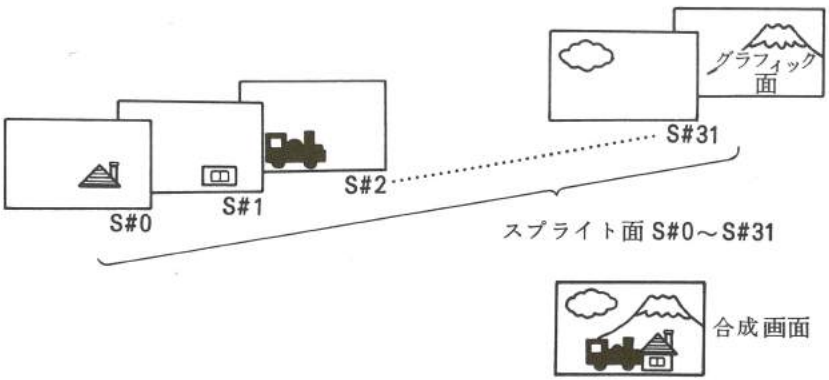
○スプライト名称

PATTERN文で付けた　S#名称のこと。MAG 1、MAG 3のように、4つのパターンを組み合せたものを使う場合は、左上のパターン、つまり、0、4、……252を指定。

PATTERN文で描いた絵にすき間があると、下の絵が見えます。このことを利用して、奥行(おくゆき)のある立体感(りったいかん)のある絵を作ることができます。

※スプライトは、水平線上に最大4パターンまで表示。4個以上のパターンが並んだ場合、優先順位の高い4個が表示。

※スプライトは、テキスト画面には使えません。

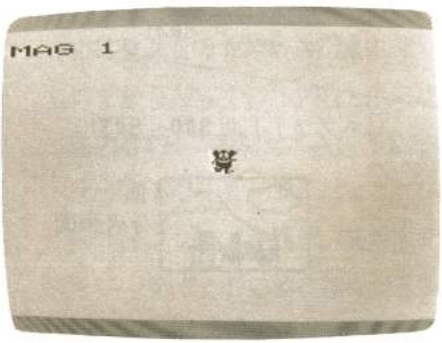


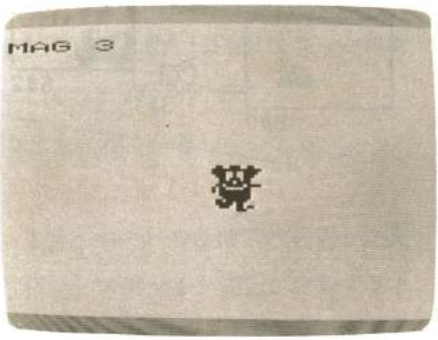
では、スプライトを使ったサンプル・プログラムを1つ載せておきましょう。

```
10 M=1
20 SCREEN 2,2:CLS
30 MAG M:C=RND(1)*13+1
40 CURSOR 10,10:PRINT CHR$(17);"MAG";M
50 FOR Y=0 TO 191 STEP 4
60 PATTERN S#0,"00193F3C1C0D0F7B"
70 PATTERN S#1,"0C0F0F0F07031B07"
80 PATTERN S#2,"00CCFE9E9CD878EC"
90 PATTERN S#3,"1AFAF8F0EC7C3800"
100 Y1=Y:GOSUB 200
110 PATTERN S#0,"00193F3C1C0D0F1B"
120 PATTERN S#1,"2C2F0F071B1F0E00"
130 PATTERN S#2,"00CCFF9E9CD878EF"
140 PATTERN S#3,"18F8F8F8F0606C70"
150 Y1=Y+2:GOSUB 200
160 NEXT Y
170 M=M+2:IF M>3 THEN M=1
180 GOTO 20
200 SPRITE 0,(120,Y1),0,C
210 SPRITE 0,(120,Y1+1),0,C
220 RETURN
```

RUNするとパターンが上から下に移動しますね。

※このように2つのパターンを縦に組み合わせて使う場合、行番号200、210のようなY座標の取り方をします。右のパターンのY座標は、左のパターンのY座標に1をたすというのは形式的なものです。





第10章　その他コンピュータの機能をフルに活用するための命令

1．音を出す

SOUND（サウンド）

SC—3000には、シンセサイザ機能があります。この機能を使えば、音楽を演奏したり、ゲーム等の効果音を出すことができます。その命令は、

SOUND　チャンネル，周波数，音量

で、チャンネル、周波数、音量は、それぞれ次のように指定します。

①チャンネル

0～5までの6個のチャンネルがあります。1つのチャンネルからは1つの音しか出ません。3重和音まで出すことができます。

0：音を消す（例：SOUND　0）

1～3：110Hz～の音を出す
4：ホワイトノイズの選択
5：同期ノイズの選択
} 主に効果音に使われる

②周波数

チャンネル１～３の場合：周波数（単位は Hz）を書く

チャンネル４、５の場合：０～２……三段階の決定されてる周波数になる。

③音量　３……チャンネル３で指定する周波数になる

０：音を消す

１：最小の音量

〜

15：最大の音量

次の表は、音階と周波数の対応(たいおうひょう)表です。メロディを奏でる時の目安にして下さい。

音階			f1	f2	f3	f4	f5
C	ド	ハ		131	262	523	1047
C♯，D♭				139	277	554	1109
D	レ	ニ		147	294	587	1175
D♯，E♭				156	311	622	1245
E	ミ	ホ		165	330	659	1319
F	ファ	ヘ		175	349	698	1397
F♯，G♭				185	370	740	1480
G	ソ	ト		196	392	784	1568
G♯，A♭				208	415	831	1661
A	ラ	イ	110	220	440	880	1760
A♯，B♭			117	233	466	932	
B	シ	ロ	123	247	494	988	

周波数単位．Hz.

BEEP（ビープ）

プログラムの中で、短い音を出したいときに使います。

BEEP　　ピッと音が鳴る。

BEEP 0　BEEP 音をとめる。

BEEP 1　ピーと鳴り続ける。

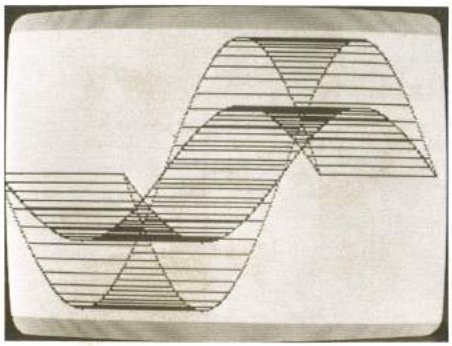
BEEP 2　ピポピポと鳴る。

2．計算機能を高める関数

コンピュータの計算機能を高めるため、SC—3000には、三角関数をはじめ、種々の関数が組み込まれています。

形　式	機　　能
ABS (X)	式Xの絶対値を与える
RAD (X)	角度からラジアンに変換
DEG (X)	ラジアンから度に変換
PI	円周率を与える
SIN (X)	三角関数の正弦の値を与える（X：ラジアン）
COS (X)	三角関数の余弦の値を与える（X：ラジアン）
TAN (X)	三角関数の正接の値を与える（X：ラジアン）
ASN (X)	三角関数の逆正弦（SINθのθの値（度数））を与える（X：－1～1）
ACS (X)	三角関数の逆余弦 COSθのθの値（度数））を与える（X：－1～1）
ATN (X)	三角関数の逆正接（TANθのθの値（度数））を与える（X：$-\frac{\pi}{2}$～$\frac{\pi}{2}$）
SGN (X)	数値Xの符号を求める（Xの値が負の時は－1、0の時は0、正の時は1）
LOG (X)	eを底とする対数を求める　　（自然対数）
LTW (X)	2を底とする対数を求める
LGT (X)	1 ϕ を底とする対数を求める　　（常用対数）
EXP (X)	自然対数の底eのべき乗を求める
SQR (X)	数値Xの両方根を求める
HEX$ (X)	10進数の数値Xを16進に変換（X：－32768～32767）

例1）　SINカーブで波模様を描く



プログラムは次のようになります。

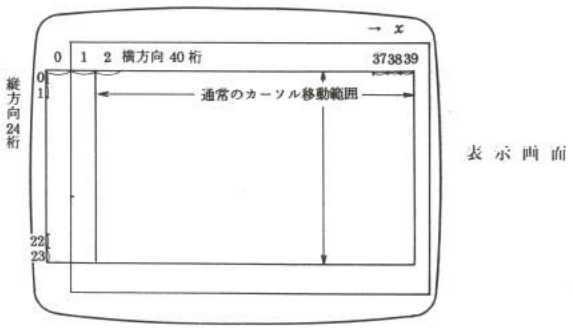
```
10 SCREEN 2,2:CLS
20 R=45
30 FOR I=0 TO 2*PI STEP PI/4/R
40 Y=R*SIN(I):PSET(X,Y+95):PSET(X+R*1.
5,Y+95):PSET(X,2*Y+95):PSET(X+R*1.5,2*
Y+95)
50 X=X+0.5:NEXT I
60 FOR J=0 TO 2*PI STEP PI/R*2
70 YY=R*SIN(J):LINE(XX,2*YY+95)-(XX+R*
1.5,2*YY+95)
80 LINE (XX,YY+95)-(XX+R*1.5,YY+95)
90 XX=XX+4:NEXT J
100 GOTO 100
```

3．VRAMに直接作用する命令

VPOKE（ブイポーク）

コンピュータの内部には、プログラムやデータ等を記憶する装置がいくつかありますが、そのひとつ、「VRAM（ビデオラム）」は、画面に表示しているデータを記憶するものです。ですから、この VRAM に、直接データを書き込んで（記憶させて）しまえば、PRINT 文や LINE 文など使わなくても、画面に文字や記号などを表示できるのです。VPOKE は、VRAM にデータを書き込む命令で、

VPOKE　アドレス，データ

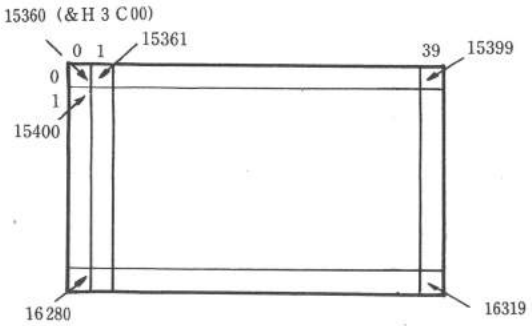


という形で使います。

VPOKEを使ってテキスト画面に表示させようとすると、いままでテキスト画面は38×24桁であったのに対し、前ページの図のように横方向に2桁増えて、40×24桁の画面を使うことができます。その画面に対応するアドレスは、左上を15360として、下図のように対応します。つまり、この40×24桁のテキスト画面の場合、アドレスは15360～16319で、座標（X,Y）（X：0～39、Y：0～23）のアドレスは、

X+Y×40+15360

という計算で求めることができます。また、VPOKE命令で使うデータは、キャラクターコードのことです。



では、次のようにダイレクト命令してみましょう。

VPOKE 15401, 65で、いままでテキスト画面では表示することができなかった(1, 1) (40×23桁の画面での座標)に、キャラクターコードが65である〝A〟が表示されましたね。

グラフィック画面にもVPOKEでドットを表示させることもできますが、グラフィック画面はテキスト画面のように単純ではないので、説明は別の機会にします。

VPEEK(ブイピーク)

VPOKEは、VRAMに書き込む命令であるのに対し、VPEEKは、VRAM内の内容を読み取ることができます。

VPEEK(アドレス)

という形で命令すれば、指定したアドレスのデータを与えてくれます。ですから、画面に描かれているキャラクター等を読み取ることもできるので、ドライブゲームやウォーゲームなどのゲームで、障害物(しょうがいぶつ)に衝突(しょうとつ)したかどうか、VPEEKで判定することができます。テキスト画面(40×24桁)の場合、アドレスはVPOKEのときと同様に計算できます。たとえば、座標(5, 20)のアドレスは、5+20×40+15360で16165となるので、

?VPEEK(16165)

で、現在の画面で(5, 20)に表示されているキャラクターのキャラクターコードを得ることができます。

4．プリンタ制御命令

LLIST（エル・リスト）

プリンタにプログラム・リストを書かせる命令はLLISTで、使い方はLISTと同じです。

○ LLIST　　：プログラムの全リストを書く
○ LLIST　行番号　　：指定した行番号のリストを書く
○ LLIST　行番号—行番号：指定した行番号から行番号までのリストを書く
○ LLIST　—行番号　　：先頭から指定した行番号までのリストを書く
○ LLIST　行番号—　　：指定した行番号以降のリストを書く

LPRINT（エル・プリント）

画面に表示する命令は、PRINTでしたが、プリンタ用紙に印字する命令は、LPRINTです。使い方はPRINT文と同様で、変数を表示させるのなら、

```
LPRINT  A
LPRINT  A$
```

といった具合に、また文字列を表示させるのでしたら、

```
LPRINT "SC—3000"
```

というようにします。

では、乱数を使ってたし算の問題を作り、それをプリンタ用紙に印字して、たし算のテストを作成するプログラムを作ってみましょう。ここで紹介するのは1～99までの数を組み合わせたたし算の問題です。プリンタを本体に接続し、プリンタの電源をONにして、このプログラムを実行して下さい。

```
10 LPRINT"***  ﾓﾝﾀﾞｲ ｸﾗｼｻﾞﾝ  ***"
20 FOR I=1 TO 10
30 A=INT(RND(1)*99)+1:B=INT(RND(1)*99)+1
40 LPRINT I;SPC(3);A;"+";B;"="
50 NEXT I
```

```
***   モンダイ < タシザン >   ***
 1      49+ 48=
 2      45+ 37=
 3      4+ 50=
 4      86+ 91=
 5      31+ 13=
 6      14+ 13=
 7      76+ 54=
 8      1+ 83=
 9      57+ 39=
 10      76+ 79=
```

HCOPY（ハードコピー）

HCOPY 命令で、いま、画面に表示されている内容を、そのまま、プリンタ用紙にコピーすることができます。

ですから、画面に手紙や絵、図表をカーソルキーを使って描いて、それをコピーすることも出来るのです。その場合、ダイレクト 命令で使い、HCOPY という文字も印字されてしまうので、画面の上の部分にコピーしたいものを 書き、カーソルを下の方へ移動し、HCOPY とダイレクト命令をするといいでしょう。そうすれば、コピーした用紙の HCOPY と 印字されている下の部分を切り取ってしまえばよいからです。

〈画面〉

```
オタンジョウビ  オメデトウ

    キョウ カラ 14 サイ

          モット BASIC デ

              ガンバリマショウ

  キミ ノ トモダチ ヨリ
```

〈プリント・アウト〉

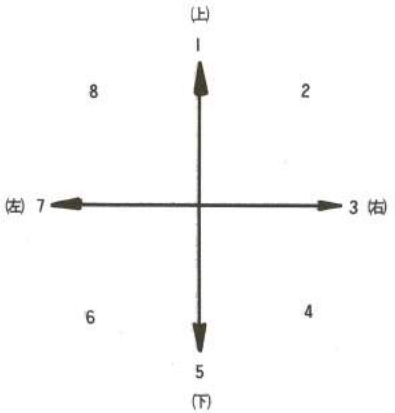
5．ジョイスティックの命令

STICK（スティック）

STICK（n）　　n＝　1：ジョイスティック1

2：ジョイスティック2

は、ジョイスティックを動かした方向を、数値で与えてくれる関数です。その値は、1～8で、ジョイスティックの方向と次のように対応します。



ですから、例えばSTICK(1)の値が5の場合、ジョイスティック1が下を向いていることを示します。

STRIG（ストリガ）

ジョイスティックのトリガのON、OFFは、

STRIG（n）　　n＝　1：ジョイスティック1

2：ジョイスティック2

という関数で知ることができます。この関数のとる値は0～3で、それぞれ次のように対応します。

0：左右のトリガ OFF

1：右のトリガ　ON、左のトリガ　OFF

2：右のトリガ　OFF、左のトリガ　ON

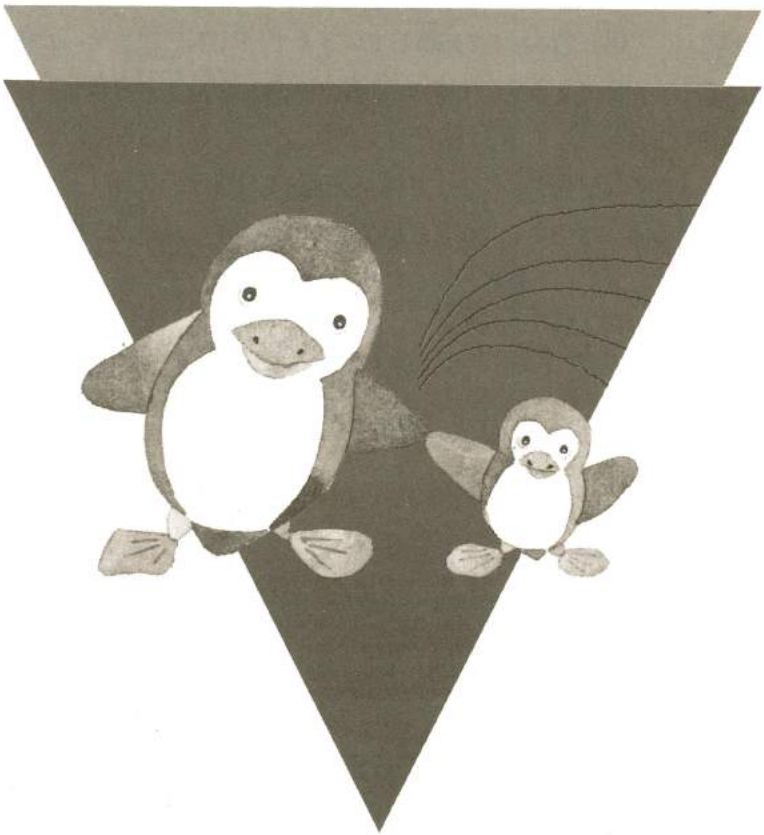
3：左右のトリガ　ON

次のプログラムは、接続したジョイスティックが正しく作動(さどう)しているかを調べるものです。画面上で、ジョイスティック1は〝1〟、ジョイスティック2は〝2〟が、スティックを動かした方向にそれぞれ移動し、トリガを押すと、左のトリガであれば数字の左に、右のトリガであれば数字の右に〝*〟が表示されれば、ジョイスティックが正しく作動していることになります。

```
20 X(1)=9:Y(1)=11:X(2)=27:Y(2)=11
30 L$=" ":R$=" "
40 FOR I=1 TO 2
50 CURSOR X(I)-1,Y(I):PRINT SPC(3)
60 P(I)=STICK(I):S(I)=STRIG(I)
70 IF P(I)>=2 AND P(I)<=4 THEN X(I)=X(
I)+1
80 IF P(I)>=6 AND P(I)<=8 THEN X(I)=X(
I)-1
90 IF P(I)>=4 AND P(I)<=6 THEN Y(I)=Y(
I)+1
100 IF P(I)=8 OR P(I)=1 OR P(I)=2 THEN
 Y(I)=Y(I)-1
110 IF X(I)<0 THEN X(I)=0
120 IF X(I)>37 THEN X(I)=37
130 IF Y(I)<0 THEN Y(I)=0
140 IF Y(I)>23 THEN Y(I)=23
150 IF S(I)=1 OR S(I)=3 THEN L$="*"
160 IF S(I)=2 OR S(I)=3 THEN R$="*"
170 CURSOR X(I)-1,Y(I):PRINT L$+STR$(I
)+R$
180 NEXT I
190 GOTO 30
```

ゲームからビジネスまで

おもしろソフト プログラム集



1. ゲーム

ブロックくずしゲーム

TVゲームの中で最(もっと)もポピュラーなゲームのひとつ、「ブロックくずしゲーム」をSC-3000でやってみることにしましょう。

遊び方、ルールはTVゲームの場合とほとんど同じで、ボールをラケットではね返し、ブロックにあてて破壊(はかい)していきます。RUN CR でゲーム開始となり、ボールが出てきます。そのボールを、←キー、→キーでラケットを左右に操作してはね返して下さい。ボールがブロックにあたるとそのブロックは消え、1つのブロックにつき1点得点が追加されます。ラケットでボールを打ち損じると、次のボールがまた出て来ますが、ボールが出て来る場所は毎回違います。また、5回打ち損じたり、44個のブロックを全て破壊してしまった場合、ゲーム終了となります。ゲーム終了時には、スコアとハイスコアが表示され、「モウイチド　ヤリマスカ？」と尋(たず)ねてくるので再プレイしたければY CR、したくなければN CR とインプットして下さい。

```
10 SCREEN 1,1
15 COLOR 10,1
20 CLS:T=5:SC=0
30 CURSOR 0,0:PRINT"** ﾌﾞﾛｯｸ ｸｽﾞｼ ｹﾞｰﾑ
 **  HIGH SCORE:";:PRINT HS
40 CURSOR 21,1:PRINT"SCORE:";:GOSUB 86
0
50 FOR I=2 TO 21:CURSOR 0,I:PRINT"X"
60 CURSOR 35,I:PRINT"X":NEXT I
70 CURSOR 1,2:FOR I=1 TO 34:PRINT"X";:
NEXT I
80 FOR Y=4 TO 10 STEP 2
90 CURSOR 2,Y:FOR I=1 TO 11:PRINT"■■■ "
;:NEXT I,Y
100 A=16:CURSOR A,22:PRINT"―"
110 XX=INT(RND(1)*13)+10:YY=11:D=INT(R
ND(1)*2)+2
120 BEEP 1:FOR I=1 TO 150:NEXT I:BEEP
0
```

初期画面を描く

ボールの出てくる位置を決める

```
130 FOR I=1 TO 300:NEXT I
140 A$=INKEY$
150 IF A$=CHR$(28) THEN 190
160 IF A$<>CHR$(29) THEN 210
170 IF A<1 THEN 210
180 A=A-2:CURSOR A,22:PRINT"──   ":GOTO
 210
190 IF A>31 THEN 210
200 A=A+2:CURSOR A-2,22:PRINT"   ──"
210 ON D GOTO 220,270,330,390
220 PP=VPEEK((YY-1)*40+XX+1+2+&H3C00):
GOSUB 810
230 ON P GOTO 240,480,600
240 GOSUB 780
250 XX=XX+1:YY=YY-1:GOSUB 790
260 GOTO 140
270 PP=VPEEK((YY+1)*40+XX+1+2+&H3C00):
GOSUB 810
280 ON P GOTO 290,500,620
290 GOSUB 780
300 XX=XX+1:YY=YY+1:GOSUB 790
310 IF YY=22 THEN 440
320 GOTO 140
330 PP=VPEEK((YY+1)*40+XX-1+2+&H3C00):
GOSUB 810
340 ON P GOTO 350,520,640
350 GOSUB 780
360 XX=XX-1:YY=YY+1:GOSUB 790
370 IF YY=22 THEN 440
380 GOTO 140
390 PP=VPEEK((YY-1)*40+XX-1+2+&H3C00):
GOSUB 810
400 ON P GOTO 410,540,660
410 GOSUB 780
420 XX=XX-1:YY=YY-1:GOSUB 790
430 GOTO 140
440 GOSUB 780:BEEP 2
450 FOR I=1 TO 300:NEXT I
460 T=T-1:IF T=0 THEN 690
470 GOTO 110
480 X=XX:Y=YY-1:GOSUB 760
490 X=XX+1:D=2:GOTO 560
500 X=XX:Y=YY+1:GOSUB 760
510 X=XX+1:D=1:GOTO 560
520 X=XX-2:Y=YY+1:GOSUB 760
530 X=XX-1:D=4:GOTO 560
540 X=XX-2:Y=YY-1:GOSUB 760
550 X=XX-1:D=3
560 GOSUB 790
570 SC=SC+1:GOSUB 800
580 IF SC=44 THEN 690
590 GOTO 140
600 IF XX=34 THEN D=4:GOTO 680
610 D=2:GOTO 680
620 IF XX=34 THEN D=3:GOTO 680
630 D=1:GOTO 680
640 IF XX=1 THEN D=2:GOTO 680
```

ラケット移動

右上にボールを動かす

右下にボールを動かす

左下にボールを動かす

左上にボールを動かす

ラケットでボールを打ち損じる

ボールがブロックに衝突

ボールが壁またはラケットにぶつかる

```
650 D=4:GOTO 680
660 IF XX=1 THEN D=1:GOTO 680
670 D=3
680 BEEP:GOTO 140
690 FOR I=1 TO 200:BEEP 1:BEEP 0:NEXT
I
700 CURSOR 1,22:PRINTSPC(30)
710 CURSOR 5,22:INPUT "ﾓｳｲﾁﾄﾞ ﾔﾘﾏｽｶ [Y
/N]? ";B$
720 IF B$="N" THEN END
730 IF B$<>"Y" THEN BEEP 2:GOTO 700
740 IF SC>HS THEN HS=SC
750 GOTO 20
760 CURSOR X,Y:PRINT SPC(3):GOSUB 780
770 BEEP:RETURN
780 CURSOR XX,YY:PRINT" ":RETURN
790 CURSOR XX,YY:PRINT"●":RETURN
800 CURSOR 27,1:PRINT SC:RETURN
810 P$=CHR$(PP)
820 IF P$=" " THEN P=1:GOTO 850
830 IF P$="■" THEN P=2:GOTO 850
840 P=3
850 RETURN
860 CURSOR 27,1:PRINT SC:RETURN
```

ゲーム終了時の処理
ブロック消す
ボール消す
ボール描く
スコア表示
ボール移動先の障害物の判定

UFO撃墜(げきつい)ゲーム

上空を左から右へ、上下にユラユラ動きながら飛んでいくUFOを、ミサイルで打ち落として下さい。

RUN CR でゲーム開始となります。上空左からUFOが次から次へと飛んで来ますので、ミサイル発射台(はっしゃだい)をカーソルキーで操作して、UFOを撃墜します。ミサイル発射台は←キーで左へ、→キーで右へ移動し、↑キーでミサイル発射となります。ミサイルの数は30個で、残りのミサイルの量は、画面左の棒グラフからわかります。また、画面右のSCOREのところには、UFO1台命中ごとに、1台UFOが描かれます。ミサイルを全部使い果たした時点でゲーム・オーバーとなり、スコア、ハイスコアが表示されます。スコアは(撃墜したUFOの台数)×10という計算です。再プレイをしたいときには、スペースキーを押して下さい。

```
10 REM ***** WAR GAME *****
20 SCREEN 2,2:MAG 1
30 GOSUB 130
40 IF MN=0 THEN 530
50 K$=INKEY$
```

初期設定
ゲーム・オーバーの判定

```
60 IF K$=CHR$(28) AND MH+16<215 THEN M
H=MH+8:GOSUB 610
70 IF K$=CHR$(29) AND MH-1>43 THEN MH=
MH-8:GOSUB 610
80 GOSUB 320
90 FX=FX+SP:GOSUB 660
100 IF FX>235 THEN FX=0
110 IF K$=CHR$(30) THEN GOSUB 350
120 GOTO 40
130 REM ::::: INITIALIZE :::::
140 CLS
150 SC=0:MN=30:SX=210:SY=85
160 PATTERN S#0,"0303030303030303"
170 PATTERN S#1,"070F1F3F7FFF3C3C"
180 PATTERN S#2,"8080808080808080"
190 PATTERN S#3,"C0E0F0F8FCFE7878"
200 PATTERN S#4,"0000000000000307"
210 PATTERN S#5,"0F0F7FFF9999FF38"
220 PATTERN S#6,"00000000000C0E0"
230 PATTERN S#7,"F0F0FEFF9999FF1C"
240 LINE(43,70)-(43,189)
250 LINE(215,70)-(215,189)
260 MH=101:GOSUB 610
270 CURSOR 218,75:PRINT"SCORE"
280 CURSOR 12,75:PRINT"POWER":CURSOR 1
4,83:PRINT"ノコリ"
290 LINE(25,93)-(27,93+MN*3),,BF
300 RETURN
310 REM ::::: START POSSITION <UFO>:::
::
320 FY=INT(RND(1)*55)
330 SP=INT(RND(1)*15)+5
340 RETURN
350 REM ::::: FIRING <ミサイル> :::::
360 MN=MN-1
370 BLINE(25,93+(MN+1)*3)-(27,93+MN*3)
,,BF
380 LINE(MH+7,170)-(MH+7,0)
390 BEEP:FOR I=1 TO 300:NEXT I
400 BLINE(MH+7,170)-(MH+7,0)
410 IF MH+7<FX OR MH+7>FX+15 THEN RETU
RN
420 BEEP
430 LINE(FX,FY+5)-(FX+15,FY+20)
440 LINE(FX,FY+20)-(FX+15,FY+5)
450 FOR I=1 TO 300:NEXT I
460 BLINE(FX,FY+5)-(FX+15,FY+20)
470 BLINE(FX,FY+20)-(FX+15,FY+5)
480 GOSUB 710
490 SX=SX+10:IF SX=250 THEN SX=220:SY=
SY+10
500 CURSOR SX,SY:PRINT"　"
510 SC=SC+10:FX=0
520 RETURN
530 REM ::::: GAME OVER :::::
540 CURSOR 70,80:PRINT"*** GAME OVER *
**"
```

ミサイル発射台の移動

UFOの移動

ミサイル発射の判定

変数の初期値代入

パターンの設定

画面初期化

UFOの速度・高さを決定

ミサイル発射

UFOに命中したか判定

UFOにミサイル命中

```
550 CURSOR 75,100:PRINT"YOUR SCORE : "
;SC
560 CURSOR 75,110:PRINT"HIGH SCORE : "
;HS
570 CURSOR 60,130:PRINT "ｽﾍﾟｰｽ ｷｰ ﾗ ｵｼ
ﾃｸﾀﾞｻｲ !!"
580 IF INKEY$<>" " THEN 580
590 IF SC>HS THEN HS=SC
600 FX=0:GOTO 30
610 REM ::::: MOVE <ﾐｻｲﾙﾀﾞｲ> :::::
620 SPRITE 0,(MH,172),0,1
630 SPRITE 0,(MH,172),0,1
640 SPRITE 0,(MH,173),0,1
650 RETURN
660 REM ::::: MOVE <UFO> :::::
670 SPRITE 1,(FX,FY),4,1
680 SPRITE 1,(FX,FY),4,1
690 SPRITE 1,(FX,FY+1),4,1
700 RETURN
710 REM ::::: ｹｽ <UFO> :::::
720 SPRITE 1,(FX,FY),4,0
730 SPRITE 1,(FX,FY),4,0
740 SPRITE 1,(FX,FY+1),4,0
750 RETURN
```

ゲーム・オーバーの処理

ミサイル発射台の表示

UFOの表示

UFOを消す

SEA BATTLE

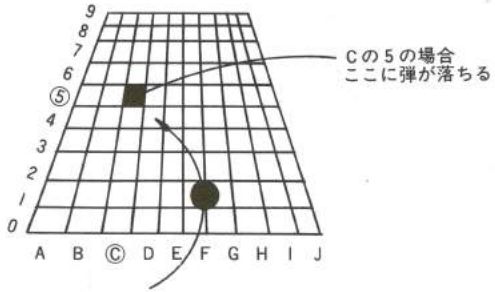
海に浮かぶ敵の5隻(せき)の船めがけて、弾を発射し、沈没させるゲームです。RUN CR でゲームスタートとなり、画面の左には海に浮かぶ5隻の船が浮いている様子、右には弾の発射目標位置を決めるための表が表示されます。左の海と、右の表とは、下のイラストのように対応しています。まず、どの船をねらうか決め、弾の発射目標位置を右の表で決めて下さい。まずコンピュータがどこに弾を発射するかを質問してきますから、0～9、A～Jで答え

て下さい。例えば、Cの5であれば、

Xホウコウ（A～J）　？　C CR

Yホウコウ（0～9）　？　5 CR

と入力します。すると「ヨロシイデスカ？」と尋ねられるのでよければY CR、よくなければN CR と入力してもう一度入力し直して下さい。入力が終ると、弾が発射される前に船が少しずつ動きます。この船が動いてしまうところがこのゲームの面白い点です。実はどのように船が移動するか予想して、弾を発射する位置を決めて下さい。1隻命中するごと得点は20点増えます。また5隻全部沈没させると、画面は描き変わり、新らたに5隻現われ、今度の場合は前回よりも1隻ごとの得点は高くなります。弾の数は、画面左のグラフでわかりますが、5隻沈没し、画面が変わるごとに弾は少し増えます。ゲームオーバーとなるのは、弾がなくなってしまった時です。




```
1 REM  ********************
2 REM  *                  *
3 REM  * SEA BATTLE GAME  *
4 REM  *                  *
5 REM  ********************
6 REM
7 REM  ムズ" カシカッタラ 1020 キ" ョウヲ
8 REM トッテクダ" サイ。
9 REM
10 DIM AD(4,9),SH(14)
100 SCREEN 2,2:HI=0
110 COLOR 15,1,,1:CLS
120 GOSUB 3010
```

```
130 CURSOR 190,100:PRINT"HI-SCORE"
140 CURSOR 230,110:PRINTHI
150 CURSOR 190,130:PRINT"    SCORE"
160 CURSOR 230,140:PRINTSC
170 REM ｻﾌﾞ ﾁｽﾞ
180 FOR X=152 TO 232 STEP 8
190 LINE(X,0)-(X,80),11:NEXT X
200 FOR Y=0 TO 80 STEP 8
210 LINE(152,Y)-(232,Y),11:NEXTY
220 FOR X=152 TO 224 STEP 8
230 CURSORX+2,85:PRINTCHR$((X-152)/8+6
5):NEXT X
240 FOR Y=0 TO 72 STEP 8
250 CURSOR235,Y:PRINT9-(Y/8):NEXTY
260 REM ﾒｲﾝ ﾁｽﾞ
270 LINE(50,90)-(150,90),1
280 LINE-(200,190),1
290 LINE-(0,190),1
300 LINE-(50,90),1
310 PAINT(100,189),5
320 COLOR 15,1
330 REM ﾀﾏ ﾉ ﾊﾟﾀｰﾝ
340 PATTERN S#0,"0F1F3F7FFFFFFFFF"
350 PATTERN S#1,"FFFFFFFF7F3F1F0F"
360 PATTERN S#2,"F0F8FCFEFFFFFFFF"
370 PATTERN S#3,"FFFFFFFFFEFCF8F0"
380 PATTERN S#4,"000000301F3F3F7F"
390 PATTERN S#5,"7F7F7F3F3F1F0F03"
400 PATTERN S#6,"0000C0F0F8FCFCFE"
410 PATTERN S#7,"FEFEFEFCFCF8F0C0"
420 PATTERN S#8,"0000000001070F1F"
430 PATTERN S#9,"1F3F3F1F1F0F0701"
440 PATTERN S#10,"00000000080E0F0F8"
450 PATTERN S#11,"F8FCFCF8F8F0E080"
460 PATTERN S#12,"0000000000000000"
470 PATTERN S#13,"0000010307070301"
480 PATTERN S#14,"0000000000000000"
490 PATTERN S#15,"0000B0C0E0E0C080"
500 PATTERN S#16,"0000000000000000"
510 PATTERN S#17,"0000000000000101"
520 PATTERN S#18,"0000000000000000"
530 PATTERN S#19,"0000000000008080"
540 PATTERN S#20,"400630125859FB7B"
550 PATTERN S#21,"FF7DF5773F1F0F07"
560 PATTERN S#22,"0170062C65277EFC"
570 PATTERN S#23,"F0F7DEBCFEF0E0C0"
580 PATTERN S#24,"0C1EFF7F00000000"
590 PATTERN S#25,"0000000000000000"
600 PATTERN S#26,"0000000000000000"
610 PATTERN S#27,"0000000000000000"
800 MAG3:GOSUB 4000
1000 REM ﾒｲﾝ ﾌﾟﾛｸﾞﾗﾑ
1010 GOSUB 2000:REM Key Input
1020 GOSUB 4100:GOSUB 4000
1030 TA=TA-1:GOSUB 3500
1040 X=84:RESTORE 1500
1050 K=KX:IFKX<5 THEN K=9-K
1060 DX=SGN(KX-4.5)*AD(K-5,KY)/30
```

初期画面を描く

海の枠

砲弾のパターン

爆発パターン

船のパターン

残りの弾を見る

```
1070 DY=DY(KY)-131:FOR TH=0TO29:READY
1080 SOUND1,1300-ABS(TH-15)*40,15
1090 X=X+DX:Y=Y+DY:IF Y<0THEN Y=0
1100 SPRITE 0,(X,Y),4*INT(TH/6),15
1110 NEXT TH:SOUND0:GOSUB 4210
1120 E=0:FORI=0TO4
1130 IF SH(I)<0 THEN E=E+1
1140 NEXT I:IF E=5 THENGOSUB 3190:S1=S
1*2:GOSUB 4000:TA=TA+5
1150 IF TA=0 THEN 4500
1160 GOTO 1010
1500 DATA 113,103,93,83,74
1510 DATA 65,57,49,42,36
1520 DATA 31,26,23,20,18
1530 DATA 18,18,19,21,25
1540 DATA 29,34,40,46,54
1550 DATA 62,70,79,89,98
1800 SPRITE 0,(X,Y),20,C:BEEP
1810 FORI=0TO200:NEXTI
1820 SPRITE 0,(X,191),20,0:RETURN
2000 REM DATA INPUT
2010 BLINE (30,12)-(140,41),,BF
2020 CURSOR 30,12:PRINT"X ﾎｳｺｳ (A~J) ?
 ";
2030 GOSUB 2180
2040 K=ASC(A$):IFK>64ANDK<75THEN2060
2050 GOTO 2030
2060 KX=K-65:PRINTA$
2070 CURSOR 30,20:PRINT"Y ｷｮﾘ  (0~9) ?
";
2080 GOSUB 2180
2090 K=ASC(A$):IFK>47ANDK<58THEN2110
2100 GOTO 2080
2110 KY=K-48:PRINTKY
2120 CURSOR 30,33:PRINT"ﾖﾛｼｲﾃﾞｽｶ? (Yor
N)";
2130 GOSUB 2180
2140 PRINTA$:IF A$="N"THEN 2160
2150 RETURN
2160 BLINE (30,12)-(140,41),,BF
2170 GOTO 2000
2180 A$=INKEY$:IF A$="" THEN 2180
2190 RETURN
3000 REM ｼｮｷ ｾｯﾃｲ
3010 RESTORE 3050
3020 FORI=0 TO 9:READ DA
3030 DY(I)=DA
3040 NEXT I
3050 DATA 180,155,142,133,124,118
3060 DATA 111,105,98,94
3070 RESTORE 3120
3080 FORI=0TO4:FORJ=9TO0STEP-1
3090 READ AD(I,J)
3100 NEXT J,I
3110 TA=10:SC=0:S1=20
3120 DATA 5,5,5,6,6,6,7,7,8,8
3130 DATA 15,16,17,17,18,19,20,22,23,2
5
```

砲弾の音

砲弾を飛ばす

弾道のデータ

命中時の水煙を描く

水煙を消す

方向・距離の入力

```
3140 DATA 25,26,28,29,30,32,34,36,39,4
2
3150 DATA 35,37,39,40,43,45,48,51,55,5
9
3160 DATA 45,47,49,52,54,58,61,65,70,7
5
3170 PATTERN C#92,"0018183C3C3C3C7E"
3180 D(0)=0:D(1)=10:D(2)=-10:D(3)=1:D(
4)=-1
3190 FOR I=0TO4
3200 K=10*RND(1)+INT(10*RND(1))*10
3210 SH(I)=INT(K)
3220 NEXT I
3230 FOR I=0TO3
3240 FOR J=I+1TO4
3250 IF SH(I)=SH(J) THEN 3260
3260 NEXT J,I
3270 RETURN
3500 REM ノコリ ノ タマ
3510 BLINE(20,55)-(130,63),,BF
3520 IF TA=0 THEN RETURN
3530 CURSOR 30,55
3540 FOR I=1TOTA
3550 PRINT "▼";:NEXT I
3560 RETURN
4000 FOR I=0TO4:IF SH(I)<0 THEN 4050
4010 SX=SH(I)MOD10:K=SX
4015 IF SX<5 THEN K=9-K
4020 SY=INT(SH(I)/10)
4030 X=SGN(SX-4.5)*AD(K-5,SY)+92
4040 SPRITE I+1,(X,DY(SY)-10),24,14
4050 NEXT I
4060 RETURN
4100 FOR I=0TO4:IF SH(I)<0 THEN 4160
4110 R=INT(5*RND(1)):K=D(R)+SH(I)
4120 IF K<1ORK>99 THEN 4110
4130 IF SH(I)MOD10=0ANDR=4THEN 4110
4140 IF SH(I)MOD10=9ANDR=3THEN 4110
4150 SH(I)=K
4160 NEXT I:RETURN
4200 REM アタリ ?
4210 C=15:A=KY*10+KX
4220 X1=KX*8+152:Y1=72-KY*8
4230 LINE(X1,Y1)-(X1+7,Y1+7),7,BF
4240 FORI=0TO4:IF A=SH(I) THEN 4310
4250 NEXT I:GOSUB 1800
4260 COLOR15,1:RETURN
4300 REM アタリ
4310 C=8:SC=SC+S1
4320 BLINE(190,140)-(255,150),,BF
4330 LINE(X1,Y1)-(X1+7,Y1+7),8,BF
4340 SH(I)=-100:SPRITE I+1,(0,0),24,0
4350 CURSOR 220,140:PRINTSC
4360 COLOR15,1:GOTO 4250
4500 BLINE (50,100)-(150,150),,BF
4510 CURSOR 60,110:PRINT"TRY AGAIN ?"
4520 CURSOR 70,130:PRINT"Y or N"
```

初期値代入

残りの弾の表示

船を描く

船の移動

ハズし

命中

```
4530 FORI=1TO5:SPRITEI,(0,191),24,0
4540 NEXT I
4550 GOSUB 2180
4560 IFSC>HI THEN HI=SC
4570 IF A$="Y" THEN 110
4580 END
```

ゲーム終了時の処理

ナンバープール

ゲーム開始前に指定した数から1までの間の数の中から、コンピュータが4つの違った数を選び横に並らべます。その順番をコンピュータのヒントをもとに、10回以内で当てるというゲームです。このゲームで、いかにコンピュータと心が通じてるか、調べられるわけです。

RUN CRで、まず「リミット ハ 1カラ イクツマデ?」とコンピュータがきいてきます。4～9の整数を入力して下さい。その指定した数から1までの間の数から、コンピュータが4つの数を選び並べるわけですから、ここで大きな数を入力するほど、難かしくなります。入力して、しばらくすると画面は変わり、ゲーム開始です。左から順に1つずつ数を入力していき、4つ並らべ終わると変更するかきいてきます。よければN、変更したければYを入力し、もう一度数を入力し直して下さい。Nを押すと、画面右下に、入力した数について、数字が当たっているか、数字と場所が当たっているかを教えてくれます。これをヒントに10回以内に当てて下さい。当たったとき、または10回入力してしまったときが、ゲーム・オーバーで、コンピュータのメッセージが表示されます。

```
10 REM  ********************
30 REM *     NUMBER POOL      *
50 REM  ********************
60 REM      [ initialis ]
70 REM T=times (1~10)
80 REM N=4 numbers  A=number limit
90 REM M(N)=machine number
100 REM I(N)=imagine number
110 REM V: 0=no much  1=much
120 PATTERN C#48,"708898A8C8887000"
130 PATTERN C#55,"F8085020504040000"
140 PATTERN C#68,"F04848E84848F000"
150 PATTERN C#79,"7088888888887000"
160 REM      [ game screen ]
170 SCREEN2,2:CLS:COLOR,1,(0,0)-(255,1
91),1
```

パターンの設定
(0(ゼロ),7,8,0(オー))

```
180 RESTORE 1490:CURSOR56,0
190 FOR S=15 TO 3 STEP-1:READ U:COLOR
S:PRINT CHR$(U);:NEXT S
200 REM       * number set *
210 COLOR15:CURSOR40,48:PRINT"ﾘﾐｯﾄ ﾊ 1
 ｶﾗ ｲｸﾂﾏﾃﾞ ? (4~9) ";
220 A$=INKEY$:IF A$=""THEN 220
230 BEEP:PRINT A$:A=VAL(A$)
240 IF A<4 OR A>9 THEN BEEP:GOSUB660:G
OTO210
250 FOR N=1 TO 4:M(N)=INT(RND(1)*A)+1:
NEXT N
260 IF M(1)=M(2) OR M(1)=M(3) OR M(1)=
M(4) THEN 250
270 IF M(2)=M(3) OR M(2)=M(4) OR M(3)=
M(4) THEN 250
280 GOSUB660
290 REM       * number set ok *
300 CURSOR16,16:COLOR3:PRINT"   ┌──────
──┐"
310 L=1:FOR R=24 TO 160 STEP16
320 CURSOR16,R:PRINT L;"|* * * *|"
330 CURSOR16,R+8:PRINT"   ├──────┤"
340 L=L+1:NEXT R
350 CURSOR16,168:PRINT"10|* * * *|"
360 CURSOR16,176:PRINT"   └──────┘"
370 CURSOR120,64:COLOR10:PRINT"○･･･ﾄﾞﾚ
ｶ ｽｳｼﾞ ｶﾞ ｱﾀﾘ!"
380 CURSOR120,72:PRINT"●･･･ﾄﾞﾚｶ ｽｳｼﾞ ﾄ
"
390 CURSOR176,80:PRINT"ﾊﾞｼｮ ｶﾞ ｱﾀﾘ!"
400 CURSOR120,96:COLOR13:PRINT" 1 2 3
4 5 6 7 8 9 10 "
410 CURSOR120,104:PRINT"┌┬┬┬┬┬┬┬┬┬┬
┬┬┬┐ "
420 FOR R=112 TO 150 STEP16
430 CURSOR120,R:PRINT"|◆|◆|◆|◆|◆|◆|◆|◆|
|◆|◆| "
440 CURSOR120,R+8:PRINT"| | | | | | | |
| | | | "
450 NEXT R
460 CURSOR120,158:PRINT"|◆|◆|◆|◆|◆|◆|◆|◆|
|◆|◆|◆| "
470 CURSOR120,166:PRINT"└┴┴┴┴┴┴┴┴┴┴
┴┴┴┘ "
480 REM       [ inkey ]
490 V=0:Y=24:XX=126:YY=112
500 FOR T=1 TO 10
510 X=34
520 FOR C=X TO X+36 STEP12:CURSOR C,Y:
PRINTCHR$(8):NEXT C
530 FOR N=1 TO 4
540 CURSORX,Y:COLOR7:PRINT CHR$(8);CHR
$(29);"?"
550 CURSOR104,24:COLOR14:PRINT"ｽｳｼﾞ ｦ
ｴﾗﾝﾃﾞ!(1 ~";A;") ";
560 I$(N)=INKEY$:IF I$(N)=""THEN 560
```

リミットのメカ

問題を決める

ゲーム画面を描く

```
570 BEEP:COLOR10:PRINT I$(N):I(N)=VAL(
I$(N))
580 GOSUB 680:IF I(N)<1 OR I(N)>A THEN
 BEEP:GOTO 550
590 CURSORX,Y:COLOR14:PRINT CHR$(8);CH
R$(29);RIGHT$(STR$(I(N)),1);
600 X=X+12:NEXT N:BEEP:BEEP
610 CURSOR104,24:COLOR14:PRINT"ヘンコウ スル
 ? [Y or N] ";
620 J$=INKEY$:IF J$="Y"THEN BEEP:COLOR
10:PRINT J$:GOSUB 680:GOTO 510
630 IF J$="N"THEN BEEP:COLOR10:PRINT J
$:GOSUB 680:Y=Y+16:GOTO 710
640 GOTO 620
650 REM ** spc & wait sub routine1 **
660 FOR C=40 TO 202 STEP 6:CURSOR C,48
:PRINT CHR$(8):NEXT C:BEEP:RETURN
670 REM ** spc & wait sub routine2 **
680 FOR C=104 TO 230 STEP 6:CURSOR C,2
4:PRINT CHR$(8):NEXT C:BEEP:RETURN
690 REM ****************************
700 REM      [ judge ]
710 BEEP:BEEP:F=0:G=0:H=0
720 FOR N=1 TO 4
730 IF I(N)=M(N) THEN I(N)=0:F=F+1
740 NEXT N
750 FOR NN=1 TO 4:FOR N= 1 TO 4
760 IF I(NN)=M(N) THEN I(NN)=0:G=G+1
770 NEXT N,NN:H=F+G
780 CURSOR XX,YY:PRINT CHR$(8)
790 CURSOR XX,YY+16:PRINT CHR$(8)
800 CURSOR XX,YY+32:PRINT CHR$(8)
810 CURSOR XX,YY+48:PRINT CHR$(8)
820 COLOR 10
830 ON H GOSUB 900,910,930,960
840 ON F GOSUB 1000,1010,1030,1060
850 IF V=1 THEN GOSUB 1110:GOTO 170
860 XX=XX+12:NEXT I
870 IF I=11 THEN GOSUB 1110:GOTO 170
880 REM ***** sub routine *****
890 REM      [ judge routine ]
900 CURSOR XX,YY:PRINT"○":RETURN
910 CURSOR XX,YY:PRINT"○"
920 CURSOR XX,YY+16:PRINT"○":RETURN
930 CURSOR XX,YY:PRINT"○"
940 CURSOR XX,YY+16:PRINT"○"
950 CURSOR XX,YY+32:PRINT"○":RETURN
960 CURSOR XX,YY:PRINT"○"
970 CURSOR XX,YY+16:PRINT"○"
980 CURSOR XX,YY+32:PRINT"○"
990 CURSOR XX,YY+48:PRINT"○":RETURN
1000 CURSOR XX,YY:PRINT"●":RETURN
1010 CURSOR XX,YY:PRINT"●"
1020 CURSOR XX,YY+16:PRINT"●":RETURN
1030 CURSOR XX,YY:PRINT"●"
1040 CURSOR XX,YY+16:PRINT"●"
1050 CURSOR XX,YY+32:PRINT"●":RETURN
1060 CURSOR XX,YY:PRINT"●"
```

数字の入力

入力データの確認

文字を消す

判定

表に○を表示

表に●を表示

```
1070 CURSOR XX,YY+16:PRINT"◆"
1080 CURSOR XX,YY+32:PRINT"◆"
1090 CURSOR XX,YY+48:PRINT"◆":V=1:RETU
RN
1100 REM      [ ending screen ]
1110 SCREEN2,2:CLS:COLOR,1,(0,0)-(255,
191),1
1120 P=80:CURSOR80,8:COLOR7:PRINT"MACH
INE NUMBER"
1130 CURSORP,24:COLOR9:PRINT"┌──────────
──────┐"
1140 CURSORP,32:PRINT"▮              ▮"
1150 CURSORP,40:PRINT"▮";M(1);" ";M(2)
;" ";M(3);" ";M(4);" ▮"
1160 CURSORP,48:PRINT"▮              ▮"
1170 CURSORP,56:PRINT"└──────────────┘"
1180 IF T=11 THEN 1290
1190 RESTORE 1400:COLOR10:FOR S=64 TO
88 STEP8:READ U:CURSOR S,80:PRINT CHR$
(U):GOSUB 1360:NEXT S
1200 CURSOR96,80:PRINT T:GOSUB 1360
1210 FOR S=120 TO 160 STEP8:READ U:CUR
SOR S,80:PRINTCHR$(U):GOSUB1360:NEXTS
1220 FOR S= 64TO 216 STEP8:READ U:CURS
OR S,96:PRINT CHR$(U):GOSUB1360:NEXTS
1230 IF T=1 OR T=2 THEN RESTORE 1410:F
OR S=96 TO 136 STEP8:READ U:CURSOR S,1
20:PRINT CHR$(U):GOSUB 1360:NEXT S
1240 IF T=3 OR T=4 THEN RESTORE 1420:F
OR S=96 TO 144 STEP8:READ U:CURSOR S,1
20:PRINT CHR$(U):GOSUB 1360:NEXT S
1250 IF T=5 OR T=6 THEN RESTORE 1430:F
OR S=88 TO 160 STEP8:READ U:CURSOR S,1
20:PRINT CHR$(U):GOSUB 1360:NEXT S
1260 IF T>=7 AND T<=9 THEN RESTORE 144
0:FOR S=96 TO 152 STEP8:READ U:CURSOR
S,120:PRINT CHR$(U):GOSUB 1360:NEXT S
1270 IF T=10 THEN RESTORE 1450:FOR S=8
0 TO 160 STEP8:READ U:CURSOR S,120:PRI
NT CHR$(U):GOSUB 1360:NEXT S
1280 GOTO 1320
1290 RESTORE 1460:FOR S=64 TO 194 STEP
8:READ U:CURSOR S,80:COLOR10:PRINT CHR
$(U):GOSUB 1360:NEXT S
1300 FOR S=96 TO 152 STEP8:READ U:CURS
OR S,96:PRINTCHR$(U):GOSUB1360:NEXT S
1310 FOR S=104 TO 144STEP8:READ U:CURS
ORS,120:PRINT CHR$(U):GOSUB1360:NEXTS
1320 FOR W=0 TO 100:NEXT W
1330 RESTORE 1480:CURSOR72,160:FOR S=4
 TO 14:COLOR S:READ U:PRINT CHR$(U);:N
EXT S
1340 FOR W=0 TO 700:NEXT W:CLS:RETURN
1350 REM      [ sound & wait ]
1360 SOUND 1,3500,15:FOR W=0 TO 2:NEXT
 W:SOUND0:FOR W=0 TO 10:NEXT W:RETURN
1370 REM ***** sub routine over *****
```

ゲーム終了の画面を描く

```
1380 REM
1390 REM      [ masage data ]
1400 DATA 183,208,255,202,182,178,255,
210,195,222,206,222,184,255,201,255,18
6,186,219,255,182,222,255,220,182,175,
192,46,46,46
1410 DATA 195,221,187,178,255,33
1420 DATA 189,186,222,178,200,255,33
1430 DATA 207,189,222,207,189,222,182,
197,255,63
1440 DATA 204,194,179,192,222,214,255,
33
1450 DATA 177,204,222,197,178,255,177,
204,222,197,178
1460 DATA 183,208,255,202,255,206,222,
184,255,201,255,186,186,219,255,182,22
2,220,182,215,197,178,46,46,46
1470 DATA 192,222,210,175,255,33
1480 DATA 17,71,65,77,69,255,79,86,69,
82,16
1490 DATA 17,78,117,109,98,101,114,255
,80,111,111,108,16
1500 DATA 73,255,76,79,86,73,78,71,255
,70,79,82,69,86,69,82,255,65,75,69,77,
73,46,77
```

ゲーム終了時メッセージのデータ

2. グラフィック

お絵描き教室

プログラムを組まずに、画面にペンで描いてるような感覚で、ドットによる細かい絵を描くことができます。

RUN CR で画面はグラフィック画面になり、ペン (⊙) が表われます。ペン操作は次のようにします。

○線を描く

4つのカーソルキー ← → ↑ ↓ で左右上下にペンを動かし、線を描くことができます。斜めの線は、[@] で右上、左下、左上、右下方向にそれぞれ描けます。

○ペンを移動する。

UYTJGMNB で、8方向にペンを移動することができます。このときは、移動するだけで、何も描きません。

○線を消す

描いた線を、EWQAZXCD で、8方向にペンを動かしながら、消すことができます。ペン先が少しでもずれていると消せませんから、うまく線上にペン先を重ねて消して下さい。

○円を描く

ペン先が置かれている位置を中心として、円を描くことができます。^ を押すと、画面はテキスト画面になり、円の半径、比率、円の書き始め、書き終りの位置を順に聞いてくるので、その値を入力して下さい。それぞれの値は CIRCLE 命令のものと同じです。

○塗りつぶす

ペン先が置かれている範囲を塗りつぶします。¥を押すと、画面はテキ

スト画面になり、塗りつぶす色を尋ねてきますので、0～15の色番号で指定します。ペン先が置かれている範囲が閉じてない場合は、画面全体が塗りつぶされてしまうので、しっかり閉じた範囲を塗りつぶすようにして下さい。また、2ヶ所以上の範囲をこのようにして塗りつぶすと、隣り合った部分がおかしくなることがあるので、できるだけ、隣り合わない部分を塗りつぶしましょう。

キー操作がたくさんありますが、慣れてしまえば、簡単でわかり易いです。

```
10 SCREEN 2,2:COLOR15,1:CLS
20 X=125:Y=95
30 PATTERNS#1,"003844829282443B"
50 REM   KEY OPES. ****
70 Z$=INKEY$
80 SPRITE0,(X-3,Y-5),1,15
90 IF Z$="^" THEN GOSUB 3000
100 IF Z$="U" THEN GOSUB 2000
110 IF Z$="Y" THEN GOSUB 2110
120 IF Z$="T" THEN GOSUB 2210
130 IF Z$="J" THEN GOSUB 2310
140 IF Z$="G" THEN GOSUB 2410
150 IF Z$="M" THEN GOSUB 2510
160 IF Z$="N" THEN GOSUB 2610
170 IF Z$="B" THEN GOSUB 2710
180 IF Z$=CHR$(28) THEN GOSUB 380
190 IF Z$=CHR$(29) THEN GOSUB 420
200 IF Z$=CHR$(30) THEN GOSUB 460
210 IF Z$=CHR$(31) THEN GOSUB 500
220 IF Z$="L" THEN GOSUB 540
230 IF Z$="@" THEN GOSUB 590
240 IF Z$="]" THEN GOSUB 640
250 IF Z$=":" THEN GOSUB 690
260 IF Z$="D" THEN GOSUB 740
270 IF Z$="A" THEN GOSUB 780
280 IF Z$="W" THEN GOSUB 820
290 IF Z$="X" THEN GOSUB 860
300 IF Z$="E" THEN GOSUB 900
310 IF Z$="Q" THEN GOSUB 950
320 IF Z$="C" THEN GOSUB 1000
330 IF Z$="Z" THEN GOSUB 1050
340 IF Z$="¥" THEN GOSUB 1120
360 GOTO 70
380 PSET(X,Y),15:X=X+1
395 GOSUB 4000
400 RETURN
420 PSET(X,Y),15:X=X-1
435 GOSUB 4000
440 RETURN
460 PSET(X,Y),15:Y=Y-1
475 GOSUB 4000
```

初期設定

入力キーの判定

上下左右に線を引くサブルーチン

```
480 RETURN
500 PSET(X,Y),15:Y=Y+1
515 GOSUB 4000
520 RETURN
540 PSET(X,Y),15
545 X=X+1
546 Y=Y-1
565 GOSUB 4000
570 RETURN
590 PSET(X,Y),15
595 X=X-1
596 Y=Y-1
615 GOSUB 4000
620 RETURN
640 PSET(X,Y),15
645 X=X+1
646 Y=Y+1
665 GOSUB 4000
670 RETURN
690 PSET(X,Y),15
695 X=X-1
696 Y=Y+1
715 GOSUB 4000
720 RETURN
740 PRESET(X,Y),1:X=X+1
755 GOSUB 4000
760 RETURN
780 PRESET(X,Y),1:X=X-1
795 GOSUB 4000
800 RETURN
820 PRESET(X,Y),1:Y=Y-1
835 GOSUB 4000
840 RETURN
860 PRESET(X,Y),1:Y=Y+1
875 GOSUB 4000
880 RETURN
900 PRESET(X,Y),1
905 X=X+1
906 Y=Y-1
925 GOSUB 4000
930 RETURN
950 PRESET(X,Y),1
955 X=X-1
956 Y=Y-1
975 GOSUB 4000
980 RETURN
1000 PRESET(X,Y),1
1005 X=X+1
1006 Y=Y+1
1025 GOSUB 4000
1030 RETURN
1050 PRESET(X,Y),1
1055 X=X-1
1056 Y=Y+1
1075 GOSUB 4000
1080 RETURN
1120 PRINT CHR$(15)
```

斜めに線を引く
サブルーチン

上下左右に線を消す
サブルーチン

斜めに線を消す
サブルーチン

```
1130 CLS:CURSOR8,12:INPUT"イロ バンゴウ ?
0~15  ";C
1135 IF C<0ORC>15 THEN1130
1140 PRINT CHR$(15)
1150 PAINT(X,Y),C
1160 RETURN
2000 REM SPRITE
2010 SPRITE0,(X-3,Y-5),1,15
2015 X=X+1
2016 Y=Y-1
2035 GOSUB 4000
2040 RETURN
2110 SPRITE0,(X-3,Y-5),1,15:Y=Y-1
2135 GOSUB 4000
2140 RETURN
2210 SPRITE0,(X-3,Y-5),1,15
2215 X=X-1
2216 Y=Y-1
2235 GOSUB 4000
2240 RETURN
2300 REM SPRITE
2310 SPRITE0,(X-3,Y-5),1,15:X=X+1
2325 GOSUB 4000
2340 RETURN
2400 REM SPRITE
2410 SPRITE0,(X-3,Y-5),1,15:X=X-1
2425 GOSUB 4000
2440 RETURN
2500 REM SPRITE
2510 SPRITE0,(X-3,Y-5),1,15
2515 X=X+1
2516 Y=Y+1
2535 GOSUB 4000
2540 RETURN
2600 REM SPRITE
2610 SPRITE0,(X-3,Y-5),1,15:Y=Y+1
2635 GOSUB 4000
2640 RETURN
2700 REM SPRITE
2710 SPRITE0,(X-3,Y-5),1,15
2715 X=X-1
2716 Y=Y+1
2735 GOSUB 4000
2740 RETURN
3000 REM CIRCLE
3010 CLS:INPUT"ハンケイ =";R
3020 C=15
3030 INPUT"ヒリツ =";L
3040 INPUT"START  0 ~.9 =";S
3050 INPUT"END    0 ~ 1 =";E
3070 CIRCLE(X,Y),R,C,L,S,E,
3080 PRINT CHR$(15)
3090 RETURN
4000 IF X=256 THEN X=255
4010 IF X=-1 THEN X=0
4020 IF Y=192 THEN Y=191
4030 IF Y=-1 THEN Y=0
4040 RETURN
```

色を塗るサブルーチン

ペン先移動のサブルーチン

円を描くサブルーチン

ペン先が画面の外に出ないように修正

幾何学模様（きかがくもよう）

SC-3000は、192×256ドットのグラフィック画面で16色が使える、という優れたグラフィック機能を持っています。このグラフィック機能をフルに活かせば、細かい絵や模様などをコンピュータで描けてしまいます。

ところで、不規則な絵や模様を描く場合、どうしてもプログラムは長くなってしまいます。そこで、10行前後のプログラムで見映えのする模様の描けるという、幾何学模様を紹介しましょう。幾何学模様とは、三角形や四角形、円などを回転させたり、横にずらしていったり、また、数学的な関数（三角関数など）を使って描いた模様のことです。ここでは、３つの幾何学模様のプログラムを紹介しておきましたが、これらを応用すれば、いろいろな幾何学模様を簡単に作ることができます。

```
10 SCREEN 2,2:CLS:C=1
20 R=20:A=0
30 FOR K=R TO 191-2*R STEP 2*R
40 FOR I=R*2 TO 255 STEP R
50 CIRCLE (I-R,K+A),R,C,1,0.5+0.5/3,1
60 CIRCLE (I,K+A),R,C,1,0.5/3,0.5
70 NEXT I:C=C+1:IF C=15 THEN C=1
80 A=A+10:NEXT K
90 GOTO 20
```

初期設定（C:色, R:半径, A:中心のY座標）

1列の線を描く

```
10 SCREEN 2,2:CLS
20 R=6:P=6:N=8:L=R*P:K=2*PI/N:C=0
30 FOR F=0 TO L STEP L
40 FOR G=1 TO 191/(L*2)
50 FOR H=1 TO 255/(L*2)
60 CL=CL+1
70 FOR I=1 TO N
80 FOR J=0 TO P
90 A=R*J*COS(K*I)+L*2*H+F-L:B=R*J*SIN(
K*I)+2*L*G+F-L
100 C=R*(P-J)*COS(K*(I+1))+L*2*H+F-L:D
=R*(P-J)*SIN(K*(I+1))+L*2*G+F-L
110 LINE (A,B)-(C,D),CL
120 NEXT J,I,H,G,F
130 GOTO 130
```

CL:色

花を1つ描く

3. 音楽

SWEET MEMORYS

SC−3000で、「SWEET MEMORIES」を演奏してみましょう。

RUN CR で演奏が開始します。メロディに合わせて、画面には模様が出てきます。SC−3000の3重和音まで出せるという音楽機能をフルに活かしたコンピュータ・サウンドを楽しんで下さい。

```
10 SCREEN 2,2:CLS
20 F=1000:GOSUB 470
30 GOSUB 130:GOSUB 130
40 GOSUB 190:GOSUB 330
50 F=900:GOSUB 470
60 GOSUB 130:GOSUB 190
70 GOSUB 200:GOSUB 200
80 GOSUB 210:GOSUB 210
90 GOSUB 220:GOSUB 190
100 GOSUB 330
110 F=900:GOSUB 470
120 GOTO 10
130 GOSUB 160:GOSUB 160
140 GOSUB 170:GOSUB 170
150 GOSUB 180:RETURN
160 RESTORE 480:NN=6:GOSUB 230:RETURN
170 RESTORE 500:NN=8:GOSUB 230:RETURN
180 RESTORE 520:NN=25:GOSUB 230:RETURN
190 RESTORE 540:NN=59:GOSUB 230:RETURN
200 RESTORE 480:NN=6:GOSUB370:RETURN
210 RESTORE 500:NN=8:GOSUB370:RETURN
220 RESTORE 520:NN=25:GOSUB370:RETURN
230 FOR N=1 TO NN
240 READ F,SP
250 X=RND(1)*250:Y=RND(1)*186:C=RND(1)
*15
```

演奏・画面表示のメインプログラム

```
260 LINE(X,Y)-(X+5,Y+5),C,B
270 IF F=700RF=1400RF=280 THEN GOSUB 4
70:NEXT N:RETURN
280 FOR V=12 TO 0 STEP SP
290 SOUND1,F,12
300 SOUND2,F*2,V
310 NEXT V,N
320 RETURN
330 FOR V=11 TO 0 STEP -1
340 SOUND1,330,V
350 NEXT V
360 RETURN
370 FOR N=1 TO NN
380 READ F,SP
390 X=RND(1)*255:Y=RND(1)*191:C=RND(1)
*15
400 CIRCLE(X,Y),5,C,1
410 IF F=700RF=1400RF=280 THEN GOSUB 4
70:NEXT N:RETURN
420 FOR V=12 TO 0 STEP SP
430 SOUND1,F,12
440 SOUND2,F*2,0
450 NEXT V,N
460 RETURN
470 FOR W=0 TO F:NEXT W:RETURN
480 REM ﾅﾂｶｼｲ~
490 DATA 247,-6,277,-6,330,-6,370,-4,4
15,-6,280,0
500 REM ｽﾞｯﾄﾏｴﾆ~
510 DATA 415,-6,370,-6,330,-6,370,-6,2
77,-6,330,-6,280,0,70,0
520 REM ﾃﾞﾓ ｱﾅﾀﾖ~
530 DATA 262,-12,247,-2,247,-6,277,-6,
311,-6,330,-2,370,-6,415,-6,392,-6,392
,-2,277,-2,330,-2,277,-2,247,-2,554,-2
,494,-2,415,-6,440,-6,494,-6,494,-4,44
0,-12,70,0,415,-6,330,-6,370,-2
540 REM ｳｼﾅｱｯﾀﾞ~
550 DATA 415,-6,494,-6,392,-6,370,-4,3
30,-12,280,0,415,-6,523,-6,415,-6,370,
-4,330,-12,280,0,415,-6,523,-6,415,-6,
370,-4,330,-12,140,0,392,-6,370,-6,330
,-6,330,-6,277,-6,330,-6,494,-2,70,0,4
15,-6,494,-6,415,-6,370,-6,70,0
560 DATA 415,-6,494,-6,415,-6,370,-4,3
30,-12,280,0,415,-6,523,-6,415,-6,370,
-4,330,-12,70,0,311,-6,330,-6,523,-6,1
40,0,247,-2,494,-1,70,0,247,-12,220,-2
,440,-2,140,0,415,-2,370,-2,330,-12,33
0,-3,280,0
```

演奏・画面表示のサブルーチン

音のデータ

4. 学習

マイ算数塾

コンピュータが作る四則（たし算、ひき算、かけ算、わり算）の計算問題で、計算力のトレーニングができるというプログラムです。四則のうち、どれをやるかは選択でき、それぞれ10問構成になっています。答えが間違っていた場合は、正しい答を表示し、また、1問正解ごとに10点として、得点もつけてくれます。

RUN CR で、画面にメニューが表われます。希望の計算を番号で入力して下さい。例えば、ひき算であれば、2 CR と入力して下さい。すると、1問ずつ問題が表われますから、順に答えていきます。答え方は、答えが31と思ったら、31CR という具合いに、答えの後に必らず CR キーを押して下さい。わり算は、答えの次に、余りも入力するようになっています。余りがないときは0を入力します。答を間違えて入力してしまった場合は←で訂正して下さい。問題は、ひき算のときには答えが負になることありません。また、どの問題も暗算（あんざん）できる程度のものです。10問答え終り、Y キーを押すと、最初のメニューに戻ります。メニューで5を選ぶと、終了です。

```
10 SCREEN 2,2:CLS:Y=8:T=0:TT=0:X1=100:
X2=100:E=120
15 COLOR1,15,(0,0)-(255,191),5
20 RESTORE 80 :YY=16
30 FOR I=0 TO 5
40 READ A$
50 CURSOR64,YY:PRINTA$
60 YY=YY+16
70 NEXT I
75 COLOR1,3,(50,10)-(200,110),5
80 DATA *** ｼｿｳ ｹｲｻﾝ ***,**** ﾀｼｻﾞﾝ .......1,**** ﾋｷｻﾞﾝ .......2
,**** ｶｹｻﾞﾝ .......3,**** ﾜﾘｻﾞﾝ
.......4,**** ｵﾜﾘ .........5
90 CURSOR40,160:PRINT"ﾄﾞﾉ ｹｲｻﾝ ｦ ｼﾏｽｶ?
ﾊﾞﾝｺﾞｳ ｦ ﾄﾞｳｿﾞ"
100 ZZ$=INKEY$
110 IF ZZ$="" THEN 100
```

初期値代入

初期画面を描く

```
115 COLOR1,15,(0,0)-(255,191),5
120 ZZ=VAL(ZZ$)
130 IF ZZ<1ORZZ>5 THEN 10
140 IF ZZ=3ORZZ=4 THEN X2=10
150 TIME$="00:00:00":CLS
160 ON ZZ GOSUB 190,280,380,470,610
170 IF ZZ=5 THEN END
180 GOTO 10
190 REM ** タシザン **
200 A$="** タシザン **":GOSUB 820
210 GOSUB 790
220 C=A+B:T=T+1:Y=Y+16
230 CURSOR16,Y:PRINT T;" ";A;" +";B;"
=";
240 GOSUB 650
250 GOSUB 830
260 IF T=10 THEN 920
270 GOTO 210
280 REM ** ヒキザン **
290 A$="** ヒキザン **":GOSUB 820
300 GOSUB 790
310 IF A<=B THEN 300
320 C=A-B:T=T+1:Y=Y+16
330 CURSOR16,Y:PRINT T;" ";A;" -";B;"
=";
340 GOSUB 650
350 GOSUB 830
360 IF T=10 THEN 920
370 GOTO 300
380 REM ** カケザン **
390 A$="** カケザン **":GOSUB 820
400 GOSUB 790
410 C=A*B:T=T+1:Y=Y+16
420 CURSOR16,Y:PRINT T;" ";A;" X";B;"
=";
430 GOSUB 650
440 GOSUB 830
450 IF T=10 THEN 920
460 GOTO 400
470 REM ** ワリザン **
480 A$="** ワリザン **":GOSUB 820
490 GOSUB 790
500 IF A<B THEN 490
510 C=INT(A/B):CC=AMODB:T=T+1:Y=Y+16
520 CURSOR16,Y:PRINT T;" ";A;"÷";B;" =
";
530 GOSUB 650:D=VAL(AA$)
540 PRINT" アマリ ";:E=170
550 FOR W=0 TO 50:NEXT W
560 GOSUB 650:DD=VAL(AA$):E=120
570 FOR W=0 TO 50:NEXT W
580 GOSUB 890
590 IF T=10 THEN 920
600 GOTO 490
610 REM ** オワリ **
620 A$="***  オワリ  ***":GOSUB 820
630 CURSOR24,100:PRINT"マタ アシタ ガンバリマシ
ョウ"
```

白板にする

時間の初期化

たし算

ひき算

かけ算

わり算

終わりの画面を描く

```
640 FOR W=0 TO 1500:NEXT W:RETURN
650 AA$="":Q=0
660 Z$=INKEY$
670 PRINT CHR$(144);CHR$(29);CHR$(8);C
HR$(29);
680 IF Z$="" THEN 660
690 IF Z$=CHR$(13) THEN RETURN
700 IF Z$=CHR$(29) THEN 740
710 PRINT Z$;:AA$=AA$+Z$
720 FOR W=0 TO 50:NEXT W
730 Q=Q+1:GOTO 660
740 Q=Q-1
750 IF Q<0 THEN Q=0:CURSOR E,Y
760 PRINTCHR$(29);
770 AA$=LEFT$(AA$,Q)
780 GOTO 660
790 A=INT(RND(1)*X1):B=INT(RND(1)*X2)
800 IF A=0ORB=0ORB=1 THEN 790
810 RETURN
820 CURSOR24,8:PRINT A$:RETURN
830 D=VAL(AA$)
840 IF C<>D THEN 870
850 TT=TT+10
860 BEEP:PRINT"  OK!":RETURN
870 BEEP2:PRINT"  NO!";C;" ﾃﾞｽ"
880 RETURN
890 IF C=DANDCC=ﾚﾙ THEN 850
900 BEEP2:PRINT"  NO!";C;" ｱﾏﾘ";CC
910 RETURN
920 CURSOR16,184:PRINT TIME$;" +++ ｶｶｯ
ﾀ ｼﾞｶﾝ      ";"PUSH Y ｷｰ"
930 CURSOR120,8:PRINT"ﾄｸﾃﾝﾊ ";CHR$(17)
;TT;CHR$(16);" ﾃﾝ"
940 Z1$=INKEY$
950 IF Z1$="" THEN 940
960 IF Z1$="Y" THEN RETURN
970 GOTO 940
```

答の入力
正解の表示

問題を作る

判定

わり算の除りの表示

時間・得点の表示

Yキーの入力待ち

2次方程式グラフ

グラフ用紙に2次方程式のグラフ（放物線状のグラフ）を描いたことはありませんか。描くだけでたいへんなので、いろいろな数値を代入してたくさんのグラフを描いてみるなんていうことはしなかったと思います。パソコンならこれがラクにできますから、自然と多くの実験をするようになり、2次式グラフの性質までがわかるようになります。数学の成績がよくなりますよ。

〈操作方法〉

RUN CR でプログラムがスタートします。a,b,cの値を入力します。もし、間違って入力したときはまたaから入力し直します。aに0を入力しま

すと、その場で再びa＝？と出ますから、0でない数値を入力して下さい。

なお、a,b,cにあまり大きな数値や小さい数値は入力しないで下さい。学校のテキストなどに載っている程度の数値を使って下さい。

グラフの頂点がX軸の±60以上やy軸の±140以上では、「グラフは描けません」（カナで）と表示されます。

なお、画面はグラフィック画面で2次式グラフを表示したあと、テキスト画面に戻りますので、グラフィック画面を見るときはSHIFT＋ BAEAK キーで画面を切り替えて下さい。再びテキスト画面に戻すときも同じ操作をして下さい。

```
1 REM 2 ｼﾞ ｷｮｸｾﾝ
5 CLS
10 PRINT"******************************
20 PRINT"                2ｼﾞ ｷｮｸｾﾝ
30 PRINT"******************************
40 PRINT:PRINT"  y=a^2 + bx + c
50 PRINT:PRINT"  a, b, c, ﾉ ｱﾀｲ ﾇ ﾆｭｳﾘ
ｮｸ ｼﾃｸﾀﾞｻｲ
60 PRINT:INPUT"a= ";A:IF A=0 THEN 60
70 INPUT"b= ";B
80 INPUT"c= ";C
90 INPUT"ﾃｲｾｲ ｼﾏｽｶ ? (Yes ﾅﾗ N)";Q$
100 IF Q$="Y" THEN 60
110 REM ***************** ｹｲｻﾝ
120 E=2*A:D=B*B-4*A*C
130 IF D>=0 THEN P=(-B+SQR(D))/E:Q=(-B
-SQR(D))/E:GOTO  150
140 P=-B/E:G=SQR(-D)/(2*A)
150 X=-B/E:Y=-D/(4*A)
160 REM **************** ｶｲﾄｳ ﾋｮｳｼﾞ
170 CLS
180 PRINT"y = ";A;"x^2 + ";B;"x + ";C
190 PRINT:PRINT"======= ANSWER ========
200 PRINT:PRINT"■■■■ ｼﾞｯｺﾝ ■■■■
210 PRINT"ｺﾝ (1)= ";P
230 PRINT"ｺﾝ (2)= ";Q:GOTO 250
240 PRINT"ｼﾞｯｺﾝ ﾊ ﾅｲ"
250 PRINT:PRINT"------ ｷｮｺﾝ ------
260 IF D>=0 THEN 290
270 PRINT"ｼﾞｯｽｳﾌﾞ =";F
280 PRINT"ｷｮｽｳﾌﾞ    =";G;"i":GOTO 300
290 PRINT"ｷｮｺﾝ ﾊ ﾅｲ
300 PRINT:PRINT"<<< ﾁｮｳﾃﾝ >>>
310 PRINT"x=";X
320 PRINT"y=";Y
330 REM ************** ｸﾞﾗﾌ ﾇ ｴｶﾞｸ
340 SCREEN 2,2:CLS
350 IF ABS(X)>60 OR ABS(Y)>140 THEN 80
0
```

a,b,cの数値の入力

根と頂点の計算

根と頂点の計算結果の表示

```
360 IF ABS(X)>6 OR ABS(Y)>14 THEN L=10
:GOTO 375
370 L=1
375 POSITION(127,95),0,1
380 REM ------------x,y ｼﾞｸ ﾗ ｴｶﾞｸ
390 LINE(-80,0)-(80,0):REM------- ﾖｺ
400 LINE(0,80)-(0,-80):REM-------- ﾀﾃ
410 REM ---------- ｶｽﾞ ﾉ ｷﾆｭｳ ﾄ ﾒﾓﾘｽﾞｹ
420 CURSOR-88,-4:PRINT-10*L:CURSOR 72,
-4:PRINT10*L:REM ------ ﾖｺ
430 CURSOR -26,80:PRINT20*L:CURSOR -26
,-80:PRINT-20*L:CURSOR -14,-1:PRINT"0"
440 FOR I=-80 TO 80 STEP 8
450 LINE(I,1)-(I,-2):REM------ x ｼﾞｸ
460 NEXT
470 FOR I=80 TO -80 STEP -8
480 LINE(-2,I)-(2,I):REM------ y ｼﾞｸ
490 NEXT
500 FOR XI=-10*L TO 10*L STEP .5*L
520 Y=A*(XI*XI)+B*XI+C
530 YI=A*((XI-L)*(XI-L))+B*(XI-L)+C
540 IF Y>20*L OR Y<-20*L THEN 580
545 IF YI>20*L OR YI<-20*L THEN 590
550 IF L=10 THEN 570
560 LINE(8*XI-8,YI*4)-(8*XI,Y*4):GOTO
590
570 LINE(.8*XI-8,YI*.4)-(.8*XI,Y*.4):G
OTO 590
580 BEEP
590 NEXT
600 END
800 CURSOR -100,90:PRINT"ｸﾞﾗﾌ ﾊ ｶｸﾏｾﾝ"
:END
```

X,Y軸,目盛,目盛上の数値,2次曲線を描く

5. ホームユース&ビジネス

体重診断

コンピュータで、体重診断をしてみましょう。身長・体重を入力すると、その身長に合った体重を計算し、現在の体重と比較してコメントを表示してくれます。

RUN CR で、まず、身長150～185cmまでの基準体重表が表示されます。次に、身長、体重を入力すると、その身長の基準体重が書き出されます。そして最後にスペース・キーを押すと画面は変わり、コンピュータの診断結果、コメントが出てきます。

```
10 REM ----- ﾀｲｼﾞｭｳ ｼﾝﾀﾞﾝ -----
20 SCREEN 1,1:CLS:SCREEN 2,2:CLS:ERASE
30 RESTORE
40 X=16:Y=16:V=88:W=10:T=0:SS=0
50 DIM D(36)
60 CURSOR72,0:PRINT" ｷｿ ﾀｲｼﾞｭｳ ﾋｮｳ "
70 LINE(16,8)-(240,8),5
80 LINE(V,W)-(V,W+104),5
90 LINE(V+80,W)-(V+80,W+104),5
100 LINE(16,W+104)-(240,W+104),5
110 REM
120 FOR S=1   TO 36
130 READ D(S)
140 SS=S+149
150 CURSORX,Y:PRINTSS;" ";D(S)
160 Y=Y+8
170 IF Y=112 THEN Y=16:X=X+80
180 NEXT S
190 IF INKEY$=" " THEN 220
200 CURSOR 24,176:PRINT"ｽﾍﾟｰｽ ｷｰ ｦ ｵｼﾃ
ｸﾀﾞｻｲ。"
210 GOTO 190
220 SCREEN 1,1
230 CURSOR2,12:INPUT"ｱﾅﾀﾉ ｼﾝﾁｮｳ ﾊ ﾅﾝｾﾝ
ﾁ ﾃﾞｽｶ。 ";I
240 CURSOR31,12:PRINT"cm"
250 CURSOR2,15:INPUT"ｱﾅﾀ ﾉ ﾀｲｼﾞｭｳ ﾊ
    ";NT
260 CURSOR31,15:PRINT"Kg"
270 FOR M=0 TO 500:NEXT M
```

基礎体重表の表示

自分のデータ入力

```
280 TT=T-149
290 SCREEN 2,2
300 CURSOR24,136:PRINT"ｱﾅﾀﾉ ｷｼﾞｭﾝ ﾀｲｼﾞ
ｭｳ ﾊ。  ";D(TT);" Kg ﾃﾞｽ。"
310 TF=((D(TT)*1.2)):TS=((D(TT)*.9))
320 CURSOR24,152:PRINT"+20 % ~ -10 % ｲ
ﾅｲ ﾏﾃﾞﾉ ﾊﾝｲｶﾞ ｱﾝｾﾞﾝﾃﾞｽ。"
330 IF INKEY$=" " THEN CLS:GOTO 360
340 CURSOR24,176:PRINT"ｽﾍﾟｰｽ ｷｰ ｦ ｵｼﾃｸ
ﾀﾞｻｲ"
350 GOTO 310
360 REM
370 SCREEN 2,2
380 BLINE (0,120)-(255,191),0,BF
390 CURSOR 28,8:PRINT "● ｱﾅﾀﾞ ﾉ ｹﾞﾝｻﾞｲﾉ
ﾀｲｼﾞｭｳ ﾊ..";NT;"Kg"
400 CURSOR 28,24:PRINT"● ﾀｲｼﾞｭｳ ｼﾞｮｳｹﾞ
ﾝ.........";TF;"Kg"
410 CURSOR 28,40:PRINT"● ﾀｲｼﾞｭｳ ｶ ｹﾞﾝ.
.........";TS;"Kg"
420 IF NT>=TF THEN 510
430 IF NT<=TS THEN 580
440 CURSOR 28,56:PRINT"○ ﾃｷｾｲ ﾀｲｼﾞｭｳ ﾃ
ﾞｽ。"
450 CURSOR 28,72:PRINT"○ ﾌﾄﾗﾅｲ ﾖｳﾆ ｼﾏｼ
ｮｳ。"
460 CIRCLE(160,106),10,1,1
470 CIRCLE(160,146),10,5,3,0,1,BF
480 LINE(160,176)-(160,186),1:LINE-(15
0,186),1
490 IF INKEY$=" " THEN 20
495 CURSOR24,176:PRINT"ｽﾍﾟｰｽ ｷｰ ｦ ｵｼﾃｸ
ﾀﾞｻｲ"
500 GOTO 490
510 REM
520 CURSOR 28,56:PRINT"▲ ｵｰﾊﾞｰ ｼﾃｲﾏｽ。
ｼｮｸｼﾞｦ ﾍﾗｼ、"
530 CURSOR 28,72:PRINT"▲ ｳﾝﾄﾞｳ ｦ ﾓｯﾄ ｼ
ﾃｸﾀﾞｻｲ。 ｷｹﾝﾃﾞｽ。"
540 CIRCLE(160,106),10,1,1
550 CIRCLE(160,146),30,8,1,.25,.75,BF
560 LINE(160,172)-(160,186),1:LINE-(15
0,186),1
570 GOTO 490
580 REM
590 CURSOR 28,56:PRINT"▲ ｽｸﾅｽｷﾞﾏｽ。
ﾀﾝﾊﾟﾂｼｮｳ ｦ ｼﾎﾞｳ ｦ、"
600 CURSOR 28,72:PRINT"▲ ﾓｯﾄ ﾄｯﾃｸﾀﾞｻｲ。
 ｷｹﾝ ﾃﾞｽ。"
610 CIRCLE(160,106),10,1,1
620 LINE(160,116)-(160,186),1:LINE-(15
0,186),1
630 GOTO 490
640 REM -------DATA-------
650 DATA 45.64,45.80,46.55,47.29,48.01
,48.71,49.31,50.01,50.70,51.39,52.09
```

自分の身長を配列の身長データに合わせる

基準体重表示

体重上限(TF),下限(TS)を計算する

現在の体重，適正体重の上限・下限の表示

体重診断

適正の場合の診断出力

スペースキーで初期画面に

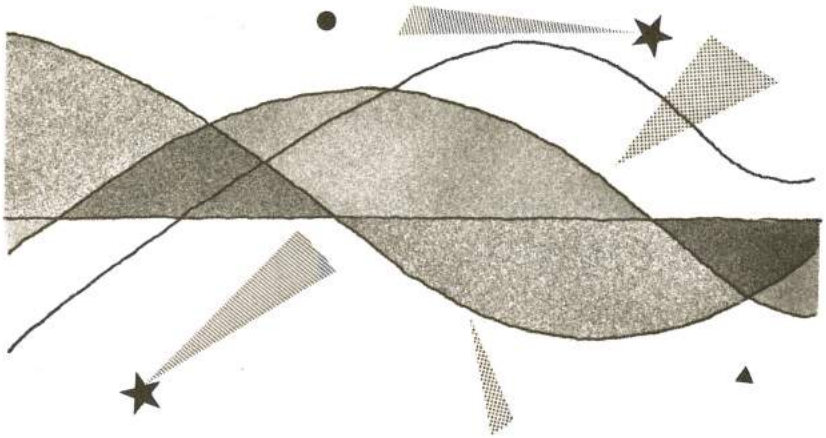
太りすぎの場合の診断出力

やせすぎの場合の診断出力

```
660 DATA 52.78,53.46,54.14,54.82,55.49
,56.19,56.87,57.54,58.20,58.86,59.51
670 DATA 60.16,60.79,61.42,62.05,62.71
,63.35,63.99,64.63,65.26,65.88,66.50
680 DATA 67.12,67.74,68.35
```

基礎体重データ

バイオリズム



生年月日と今日の年月日をインプットすると、そのデータからコンピュータが計算、バイオリズムのカーブを画面に描き、身体、感情、知性のそれぞれの注意日を教えてくれます。

RUN CR で、まず、生年月日のインプットをします。生まれた年は、西暦、または昭和、大正、明治、どちらの形でもできますから好きな方を選択して下さい。コンピュータが生年月日について順に質問してきますので、それに従ってインプットしていきます。

例) 生年月日が昭和45年1月23の場合

(1)メイジ (2)タイショウ (3)ショウワ (4)セイレキ

ウマレハショウワ ナン年？ 45 CR

ウマレハ ナンガツ？ 1 CR

ウマレハ ナンニチ？ 23 CR

次に、今日の年月日を、コンピュータが質問する順にインプットしていきま

す。ただし、今日の年月日の年は西暦で入れて下さい。全部終わると最後に「ウエノデータ デヨロシイデスカ？」と尋ねてきます、確認してよくなければ N CR とインプットして、もう1度最初からやり直します。よい場合は、Y CR とインプットすると、バイオリズムのカーブと、身体、感情、知性の注意日が表示されます。

```
10 DIM A(11),DATE$(19),MO(12),C(19)
20 RESTORE 60
30 FOR I=0 TO 11
40 READ A(I)
50 NEXT I
60 DATA 0,31,59,90,120,151,181,212,243
,273,304,334
70 NE$(1)="ﾒｲｼﾞ":NE$(2)="ﾀｲｼｮｳ":NE$(3)
="ｼｮｳﾜ":NE$(4)="ｾｲﾚｷ"
80 NE(1)=1867:NE(2)=1911:NE(3)=1925:NE
(4)=0
90 RESTORE 120
100 FOR I=1 TO 12
110 READ MO(I):NEXT I
120 DATA 31,28,31,30,31,30,31,31,30,31
,30,31
130 CLS
180 CURSOR 8,5:PRINT"BIORHYTHM(ﾊﾞｲｵﾘｽﾞ
ﾑ)"
230 CURSOR 1, 9:PRINT"**********
**********"
240 CURSOR 1,10:PRINT"*";SPC(32);"*"
250 CURSOR 1,11:PRINT"* ﾀﾝｼﾞｮｳﾋﾞ
  ｷｮｳ ﾉ ﾋｽﾞｹ       *"
260 CURSOR 1,12:PRINT"* ￥￥￥￥ｾｲ ￥￥月 ￥￥日
  ￥￥￥￥ｾｲ ￥￥月 ￥￥日  *"
270 CURSOR 1,13:PRINT"*";SPC(32);"*"
280 CURSOR 1,14:PRINT"**********
**********"
290 CURSOR 10,18:PRINT "ｱﾅﾀﾉ ﾀﾝｼﾞｮｳﾋﾞ
ｦ"
300 CURSOR 3,20:INPUT "(1)ﾒｲｼﾞ (2)ﾀｲｼｮ
ｳ (3)ｼｮｳﾜ (4)ｾｲﾚｷ";S:GOSUB 680
310 IF S<1ORS>4THEN 300
320 CURSOR 12,11:PRINTNE$(S)
330 CURSOR 5,20:PRINT"ｳﾏﾚﾀ";NE$(S);" ﾅ
ﾝﾈﾝ";:INPUT BY:GOSUB 680
340 IF BY<0 THEN 330
350 CURSOR 3,12:PRINTRIGHT$("    "+STR
$(BY),4)
360 BY=BY+NE(S)
370 CURSOR 5,20:INPUT "ｳﾏﾚﾀ ﾅﾝｶﾞﾂ    ?
";BM:GOSUB 680
380 IF (BM<1)OR(BM>12) THEN 370
390 CURSOR 9,12:PRINTRIGHT$(" "+STR$(B
M),2)
400 CURSOR 5,20:INPUT "ｳﾏﾚﾀ ﾅﾝﾆﾁ      ?
```

初期値代入

初期画面を描く

誕生年月日の入力

```
";BD:GOSUB 680
410 IF (BD<1)OR(BD>31) THEN 400
420 CURSOR13,12:PRINTRIGHT$(" "+STR$(B
D),2)
430 CURSOR 10,18:PRINT "ｷｮｳﾉ 年,月,日
?"
440 CURSOR 5,20:INPUT "ｺﾄｼﾊ ｾｲﾚｷ ﾅﾝﾈﾝ?
";TY:GOSUB 680
450 IF TY<BY THEN 440
460 CURSOR19,12:PRINTRIGHT$(" "+STR$(T
Y),4)
470 CURSOR 5,20:INPUT "ｺﾝｹﾞﾂﾊ ﾅﾝｶﾞﾂ ?
";TM:GOSUB 680
480 IF (TM<1)OR(TM>12) THEN 470
490 CURSOR25,12:PRINTRIGHT$(" "+STR$(T
M),2)
500 CURSOR 5,20:INPUT "ｷｮｳﾊ ﾅﾝﾆﾁ ?
";TD:GOSUB 680
510 IF (TD<1)OR(TD>31) THEN 500
520 CURSOR29,12:PRINTRIGHT$(" "+STR$(T
D),2)
530 CURSOR 5,16:PRINT"ｳｴﾉ ﾃﾞｰﾀ ﾃﾞﾖﾛｼｲﾃ
ﾞｽｶ ? (Y or N)"
540 GET$=INKEY$
560 IF GET$="Y" THEN 590
570 IF GET$="N" THEN 130
580 GOTO 540
590 IF (TYMOD4=0)AND(TYMOD100<>0) THEN
 MO(2)=29
600 IF (TYMOD100=0)AND(TYMOD400<>0) TH
EN MO(2)=28
610 D1=TY*365-INT((TY-1)/100)+TD+A(TM-
1)+INT((TY-1)/4)+INT((TY-1)/400)
620 D2=BY*365-INT((BY-1)/100)+BD+A(BM-
1)+INT((BY-1)/4)+INT((BY-1)/400)
630 IF MO(1)=29ANDTM>2 THEN D1=D1+1
640 DAY=D1-D2
650 CURSOR 5,18:PRINT"ｱﾅﾀﾉ ｳﾏﾚﾃｶﾗﾉ ﾆｯｽ
ｳ":DA+1:"ﾋﾒ ﾃﾞｽ "
660 GOTO 700
670 REM SPACE PRINT
680 CURSOR 0,20:PRINT SPC(37)
690 RETURN
700 SCREEN 2,1:COLOR1,15:CLS
710 LINE(60,0)-(60,140),1:LINE(60,70)-
(220,70),1
720 FOR X=68 TO 220 STEP 8
730 LINE(X,68)-(X,72),1
740 NEXT X
750 CURSOR15,160:PRINT"P･ｼﾝﾀｲ･･･ﾁｭｳｲ ﾋ
ﾞ･>"
760 CURSOR15,170:PRINT"S･ｶﾝｼﾞｮｳ･ﾁｭｳｲ ﾋ
ﾞ･>"
770 CURSOR15,180:PRINT"I･ﾁｾｲ･･･ﾁｭｳｲ ﾋ
ﾞ･>"
780 RESTORE 830
790 FOR I=0 TO 4
800 READ X,Y,AA$
```

今日の年月日の入力

入力データの確認

生まれてからの日数を計算・表示

表示を消す

テキスト画面表示

グラフィック画面に表を描く

```
810 CURSORX,Y:PRINTAA$
820 NEXT I
830 DATA 50,25,+,25,39,ｺｳﾁｮｳ,50,67,0,2
5,95,ﾃｨﾁｮｳ,50,109,-
840 FOR I=0 TO 19
850 DATE$(I)=RIGHT$(STR$(TM)+"/"+RIGHT
$(STR$(TD),2),5)
860 TD=TD+1
870 IF TD>MO(TM) THEN TD=1:TM=TM+1
880 IF TM=13 THEN TM=1:TY=TY+1
890 NEXT I
900 SCREEN 2,2
910 FOR I=DA TO DA+19
920 C(I-DA)=0
930 Y=-SIN(2*PI*(IMOD23)/23)*70+70
940 X=(I-DA)*8+60
950 IF I=DA THEN PSET(X,Y),1
960 LINE-(X,Y),1
970 IF (Y>60)AND(Y<80) THEN C(I-DA)=1
980 NEXT I
990 CURSORX,Y:PRINT"P"
1000 REM I-SIN
1010 FOR I=DA TO DA+19
1020 Y=-SIN(2*PI*(IMOD33)/33)*70+70
1030 X=(I-DA)*8+60
1040 IF I=DA THEN PSET(X,Y),1
1050 IF (Y>60)AND(Y<80) THEN C(I-DA)=C
(I-DA)+4
1060 LINE-(X,Y),1
1070 NEXT I
1080 CURSORX+6,Y:PRINT"I"
1090 REM S-SIN
1100 FOR I=DA TO DA+19
1110 Y=-SIN(2*PI*(IMOD28)/28)*70+70
1120 X=(I-DA)*8+60
1130 IF I=DA THEN PSET(X,Y),1
1140 IF (Y>60)AND(Y<80) THEN C(I-DA)=C
(I-DA)+2
1150 LINE-(X,Y),1
1160 NEXT I
1170 CURSORX+12,Y:PRINT"S"
1180 REM DATE
1190 B=4
1200 FOR Y=180 TO 160 STEP-10
1210 CURSOR133,Y
1220 FOR X=0 TO 19
1230 IF C(X)>=B THEN PRINTDATE$(X);:C(
X)=C(X)-B
1240 NEXT X
1250 B=B/2
1260 NEXT Y
1270 CURSOR80,150:PRINT TY;DATE$(0);"
~";DATE$(19)
1280 IF INKEY$="" THEN 1280
1290 SCREEN 1,1:CLS
1300 GOTO 90
```

"知性"のグラフを描く

"身体"のグラフを描く

"感性"のグラフを描く

注意日の表示

表示グラフの期間を示す

初期画面に戻るかの判定

住所録

SC-3000BASICには、DATAをカセットテープに書き込むステートメントがありません。

住所録などデータを、プログラムのDATA文に書き込めるプログラムです。

各項目のキー入力が終ると、行番号DATA文が表示されますので、そのままCRを続けて2回押してください。

これを応用して各種のデータファイルを作っておいてください。

```
10 REM ***** DATA FILES *****
20 DIM MK$(50)
30 CLS:PRINT "DATA FILE"
40 FOR J=1 TO 50:MK$(J)=""
50 CURSOR 3,2:PRINT"DIM= ";J:NEXT J
60 CURSOR 2,4:PRINT"ｼﾝｷ ﾄｳﾛｸ....1"
70 CURSOR 2,6:PRINT"ﾅﾏｴﾖ ｻｶﾞｽ...2"
80 CURSOR 2,8:PRINT"ｵﾜﾘ ........3"
90 PRINT:INPUT" ｼﾞｯｺｳ ｽﾙ No. ?";A
100 ON A GOTO 110,310,420
110 CLS:N=0:RESTORE
120 CURSOR 3,2:PRINT"DIM= ";J
130 READ IA$,IB$,IG$,IC$,ID$,IE$
140 CURSOR25,2:PRINT "ﾃﾞｰﾀｰ No.=";N
150 IF IA$<>"ｵﾜﾘ" THEN N=N+1:GOTO 130
160 CURSOR 0,13:INPUT "ﾅﾏｴ ? ";IA$
170 INPUT "ﾃﾞﾝﾜ ? ";IB$
180 INPUT "ﾕｰﾋﾞﾝNo. ? ";IG$
190 INPUT "ｼﾞｭｳｼｮ ? ";IC$
200 INPUT "ｷｰ ﾜｰﾄﾞ ? ";ID$
210 INPUT "ｵﾎﾞｴｶﾞｷ ? ";IE$
220 FOR J=3 TO 22:CURSOR 0,J:PRINT CHR
$(5);:NEXT J
230 A$="****< CR ｷｰ ﾖ 2 ｶｲ ｵｽ。>****"
240 CURSOR 2,20:PRINT A$:CURSOR 0,5
260 PRINT N+3000;" DATA ";IA$;",";IB$;
",";IG$;",";IC$;",";ID$;",";IE$
270 PRINT "GOTO 30":PRINT CHR$(10):CUR
SOR 0,4:STOP: GOTO 30
310 CLS:PRINT "ﾅﾏｴ ﾖ ｻｶﾞｽ。 ":PRINT
320 INPUT "ﾅﾏｴ ? ";SN$
330 RESTORE
340 READ IA$,IB$,IG$,IC$,ID$,IE$
350 IF SN$<>IA$ THEN 340
360 PRINT IA$:PRINTIB$:PRINTIG$:PRINTI
C$:PRINTID$:PRINTIE$
370 IF INKEY$=" " THEN 30
380 CURSOR5,20:PRINT"ｽﾍﾟｰｽ ｷｰ ﾖ ｵｽ。"
390 GOTO 370
```

初期画面

データ読み込み

データを書き込む

データを捜す

```
400 IF IA$<>"END" THEN GOTO 290
410 GOTO 30
420 CLS:PRINT" オワリ "
1000 DATA SEGA,742-3171,144,TOKYO,S,SC
-3000
9999 DATA オワリ,,,,,,
```

'オワリ'の表示

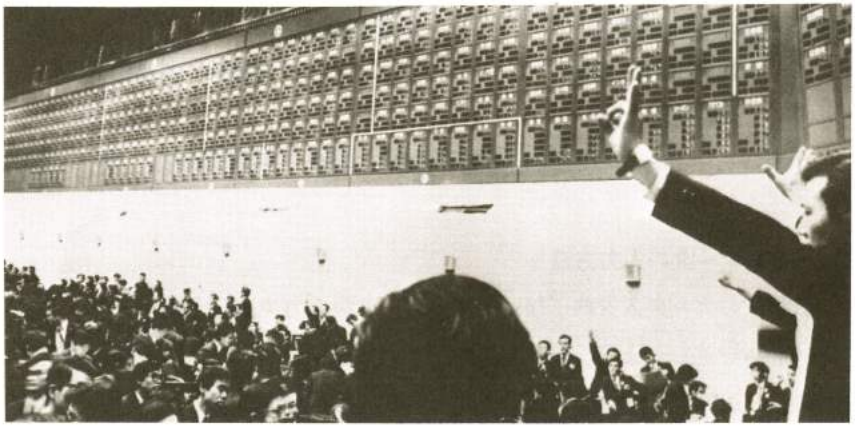
ダミーのデータ

株式チャート

株を売買するときに絶対に欠かすことのできないものの１つは、「チャート」と呼ばれる株価の推移(すいい)をグラフに表わしたものです。「チャート」と呼ばれるものにはいろいろありますが、ここにご紹介するのはもっとも一般的な「ローソク足」です。ローソク足は日本で生まれたものですが、今では世界共通の描き方になりつつあります。そして新聞や雑誌に載っているローソク足のチャートはほとんどがコンピュータによって描かれています。パソコンを使って銘柄選択(めいがらせんたく)をしたり、売買のタイミングを知り、儲(もう)けていらっしゃる方も珍しくなくなりました。

ローソク足による株価の位置の判断は株の専門書をご覧いただくことにして、ここでは簡単なローソク足の見方について触れておきましょう。

ローソク足は１日（または１週、１月……）の動きの中での始値、高値、安値および終値の４つの値を表現したものです。



〈操作方法〉

この「株式チャート」プログラムはデータ（日付や株価など）を含んでいますから、カセット・テープへは、データ込みのプログラムが記録されることになります。

出来高は、銘柄ごとの株の１日（また１週、１月……）の取引高のことです。出来高はふつう 少ない方が株価は相対的に低く、多くなってきますと株価は高くなっていきます。ローソク足はふつうは出来高といっしょに表示されることが多いので、このプログラムでも出来高を表示するようにしました。

データの書き替えはリストを映し、DATA文のみ 書き替えます。一たんDATAを入力しますと、あとはDATA文の２行（１銘柄につき）だけを書き替えればよいのです。

≪37日分のデータの入力方法≫

プログラム・リストはDATA文が書き込まれた状態になってます。最初にプログラムを入力するときは、プログラムの1000行以下にリストの例にならって、好みの銘柄のデータを入力して下さい。次に、他の銘柄のプログラムを作りたいときは、上の方法で作ったプログラムをカセット・テープに記録した後、1000行以下のデータ文のみを書き替えます。

1000行は銘柄名と銘柄コードです。銘柄コードが不要のときは銘柄名のあとの[,]の次を空らんにして下さい。

1001行以下の各行は日付と株価（始値、高値、安値、終値）と出来高です。

≪日々のデータの入力方法≫

37日分のデータ入力終了後、日々のデータを入力する場合は、1037行の次に1038行を追加します。そして1001行を削除します。その次の日は1039行を追加し、1002行を削除します。こうして順次行番号を増やしていき、9000行ぐらいになりましたら再び1000番台へ戻すようにすればよいでしょう。

≪プログラムの実行≫

キーボードまたはカセット・テープからプログラムの入力を終了しましたら、RUN CR でプログラムを実行します。最初「カブシキ　チャート」の文字が画面に現われ、２～３秒してからグラフを描きはじめます。全部描き終わるまでに20数秒かかります。

```
5 CLS
10 PRINT "****************************
20 PRINT "          ｶﾌﾞｼｷ  ﾁｬｰﾄ
30 PRINT "****************************
40 DIM A(5,50),D$(50)                          ―― 配列宣言
50 READ M$,MC$
60 FOR J=1 TO 37:READ D$(J):FOR I=1 TO
 5                                             ―― 1000行以下のDATA文の
70 READ A(I,J)                                    読み込み
80 NEXT I:NEXT J
90 SCREEN 2,2:CLS                              ―― ｸﾞﾗﾌｨｯｸ画面の指定
100 CURSOR 10,0:PRINT M$;"  ";MC$              ―― 銘柄名、銘柄コードの表示
110 HI=0
120 FOR J=1 TO 37
130 IF HI<A(2,J) THEN HI=A(2,J)                ―― 期間内高値の検出
140 NEXT J
150 LO=HI
160 FOR J=1 TO 37
170 IF A(3,J)=0 THEN 190
180 IF LO>A(3,J) THEN LO=A(3,J)                ―― 期間内安値の検出
190 NEXT J
200 CURSOR 60,10:PRINT"ﾀｶﾈ=";HI;:PRINT         ―― 期間内高値・安値の表示
"      ﾔｽﾈ=";LO
210 MAX=INT(HI/20+1)*20
220 MIN=INT(LO/20)*20                          ―― ｸﾞﾗﾌ目盛、上・下限の設定
230 Y=INT((MAX-MIN)/5+.5)                      ―― 目盛1単位の金額
240 LINE(45,31)-(200,131),2,B
250 FOR K=1 TO 5
260 LINE(45,191-(20*K+60))-(200,191-(2         ―― 外枠、目盛線を描く
0*K+60))
270 NEXT K
280 FOR K=0 TO 5
290 CURSOR 8,191-(20*K+65):PRINT MIN+Y         ―― 目盛に金額を入れる
*K
300 NEXT K
310 R=100/(MAX-MIN)                            ―― 1円当たりのドット数
320 FOR J=1 TO 37
330 IF A(1,J)=0 THEN 430
340 O=131-R*(A(1,J)-MIN)
350 C=131-R*(A(4,J)-MIN)
360 H=131-R*(A(2,J)-MIN)
370 L=131-R*(A(3,J)-MIN)
380 LINE(J*4+45,O)-(J*4+47,C),6,B              ―― ローソク足を描く
390 IF O<C THEN 420
400 LINE(J*4+46,H)-(J*4+46,C)
410 LINE(J*4+46,L)-(J*4+46,O):GOTO 430
420 LINE(J*4+46,H)-(J*4+46,L)
```

```
430 NEXT J
440 VM=0
450 FOR J=1 TO 37
460 IF VM<A(5,J) THEN VM=A(5,J)
470 NEXT J
480 LINE(45,135)-(200,175),1,B
490 CURSOR 8,135:PRINT VM
500 CURSOR 30,172:PRINT "0"
510 R=40/VM
520 FOR J=1 TO 37
530 T=INT(R*A(5,J)+.5)
540 LINE(J*4+46,175)-(J*4+46,175-T),5
550 NEXT J
560 CURSOR 24,180:PRINT D$(1)
570 FOR J=1 TO 37
580 IF RIGHT$(D$(J),1)<>"0" THEN 600
590 CURSOR J*4+40,180:PRINT RIGHT$(D$(
J),2)
600 NEXT J
610 GOTO 610
620 END
1000 DATA ニホンサンソ,(4091)
1001 DATA 11/1, 334,335,331,333,224
1002 DATA 11/2, 331,331,324,330,200
1003 DATA 11/4, 330,330,322,322,94
1004 DATA 11/5, 322,323,320,321,53
1005 DATA 11/7, 316,317,316,316,146
1006 DATA 11/8, 320,320,315,318,84
1007 DATA 11/9, 318,320,315,316,92
1008 DATA 11/10,318,318,318,318,70
1009 DATA 11/11,318,318,315,316,86
1010 DATA 11/14,326,335,323,332,353
1011 DATA 11/15,337,337,332,332,309
1012 DATA 11/16,332,333,330,333,284
1013 DATA 11/17,332,333,331,332,108
1014 DATA 11/18,330,330,320,320,157
1015 DATA 11/19,320,321,320,320,39
1016 DATA 11/21,335,335,330,331,550
1017 DATA 11/22,343,346,336,343,1071
1018 DATA 11/24,349,354,344,350,1271
1019 DATA 11/25,350,355,342,348,1436
1020 DATA 11/26,350,351,348,350,875
1021 DATA 11/28,352,356,349,354,1552
1022 DATA 11/29,357,362,356,359,3237
1023 DATA 11/30,365,368,361,363,3917
1024 DATA 12/1, 368,385,367,385,8329
1025 DATA 12/2, 394,400,388,388,9270
1026 DATA 12/3, 388,393,383,386,1384
1027 DATA 12/5, 385,387,375,378,1019
1028 DATA 12/6, 375,376,366,366,952
1029 DATA 12/7, 370,378,366,378,835
1030 DATA 12/8, 377,382,370,382,1059
1031 DATA 12/9, 382,385,375,375,945
1032 DATA 12/12,372,378,366,367,426
1033 DATA 12/13,372,372,366,366,313
1034 DATA 12/14,370,375,369,373,374
1035 DATA 12/15,385,391,377,387,2426
1036 DATA 12/16,390,402,388,392,4666
1037 DATA 12/17,405,410,400,400,3928
```

出来高最高値の検出

出来高チャートの枠を描く

出来高最高値と0の金額を表示

出来高チャートを描く

第1日目の日付の表示

0のつく日付(10,20,30日)を表示

グラフィック画面に固定

データ
(日付,始値,高値,安値,終値,出来高の順)

ローンの金利計算

借入年数や借入額、利率が同じなのに返済額はローンの計算方式によって違います。ここではアドオン、元利均等返済、元金均等返済の3方式をとりあげました。借入年数、借入額および利率を入力すると、各ローンの月々の返済額と返済額のトータルが一覧表になって表示されます。

住宅ローンなどに利用してみてはいかがですか。どのローンがトクか、よーく考えてからお金を借りましょう。

〈操作方法〉

RUN CR で初期画面が出ます。指示にしたがって借入年数、借入額、利率を入力して下さい。数秒間（借入年数によって待つ時間が異なります）たつと、比較表が出ます。元金均等返済方式の場合は、毎日の返済額が異なりますから、それを見たいときは①を押します（②なら終了）。表の下に続けて毎月の返済額と残高が表示されます。月数が多いと画面がせり上りますから、これを止めたいときは BREAK キーを押して下さい。止めたあとまた続けて見たいときは、CONT CR として下さい。

```
5 CLS
10 PRINT"****************************
20 PRINT"          ローン ケイサン
30 PRINT"****************************
40 PRINT:INPUT"ネン スウ          (ネン) =";N:P=N*
12
50 PRINT:INPUT"カリイレ ソウガク(エン)=";K:TH=K
/P
60 PRINT:INPUT"ネン リ       (%)=";NR:PR=NR/1
2/100
70 REM -------------- アドオン
80 AD=INT(K*(1+P*PR)/P+.5)
90 REM ---------------- ガンリ キントウ
100 GR=INT(K*(((1+PR)^P*PR)/((1+PR)^P-
1))+.5)
110 REM --------------- ガンキン キントウ
130 DIM GK(P),GT(P),GL(P)
140 FOR I=1 TO P
150 GK(I)=INT(TH+(K-(TH*(I-1)))*PR+.5)
160 GT=GT+GK(I)
170 GL(I)=GT
180 NEXT I
190 CLS:REM ---------- ヒカク ヒョウ
200 PRINT"ハンサイ ネンスウ =";N;"   (ツキスウ=";P;")
"
```

借入年数,借入額,年利

アドオン,元利均等返済,元金均等返済各方式の計算

```
210 PRINT"カリイレ ｶﾞｸ ="；K
220 PRINT"ネンリツ  (%)  ="；NR；：PRINT TAB(
26)；"ダンイ: エン
230 PRINT"┌──────┬──────────┬──────
──────┐
240 GOSUB 540
250 GOSUB 530
260 GOSUB 540
270 GOSUB 530
280 GOSUB 550
290 GOSUB 530
300 GOSUB 550
310 GOSUB 550
320 PRINT"└──────┴──────────┴──────
──────┘"
330 CURSOR 8,4：PRINT"1カ月ﾍﾝｻｲｷﾝｶﾞｸ"
340 CURSOR 22,4：PRINT"ソウ ﾍﾝｻｲｷﾝｶﾞｸ"
350 CURSOR 1,6：PRINT"ｱﾄﾞｵﾝ"：CURSOR 8,6
：PRINT AD：CURSOR 22,6：PRINT AD*P
360 CURSOR 1,8：PRINT"ｶﾞﾝﾘｷﾝ"：CURSOR 8,
9：PRINT GR：CURSOR 22,9：PRINTGR*P
370 CURSOR 1,9：PRINT"ﾄｳﾍﾝｻｲ
380 CURSOR 1,11：PRINT"ｶﾞﾝｷﾝ"：CURSOR8,1
1：PRINT"ﾊｼﾞﾒ ﾉ ﾂｷ"
390 CURSOR 1,12：PRINT"ｷﾝﾄｳ"：CURSOR8,12
：PRINTGK(1)：CURSOR22,12：PRINTGT
400 CURSOR 1,13：PRINT"ﾍﾝｻｲ"：CURSOR8,13
：PRINT"ｵﾜﾘ ﾉ ﾂｷ"
410 CURSOR 8,14：PRINT GK(P)
420 CURSOR 1,16：PRINT"ｶﾞﾝｷﾝ ｷﾝﾄｳ ﾍﾝｻｲ
ﾎｳｼｷ ﾉ ﾒｲｻｲ
430 CURSOR 1,17：PRINT"ﾒｲｻｲ ﾋﾂﾖｳ...1  ｲ
ﾗﾅｲ...2
440 Q$=INKEY$：IF Q$=""THEN 440
450 IF Q$="1"THEN 470
460 END
470 PRINT：PRINT"----------ｶﾞﾝｷﾝ ｷﾝﾄｳ ﾒｲｻ
ｲ
480 PRINT"ﾄｳｹﾂ ﾂｷﾍﾝｻｲｶﾞｸ  ｻﾞﾝﾀﾞｶ
490 FOR I=1 TO P
500 PRINT I；")"；GK(I)；"    "；GT-GL(I)
510 NEXT I
520 END
530 PRINT"├──────┼──────────┼──────
──────┤"：RETURN
540 PRINT"|      |          |
      |"：RETURN
550 FOR I=1 TO 2：PRINT"|      |
    |          |"：NEXT I：RETURN
```

入力データの表示

比較表の枠を描く

比較表の中をうめる

元金均等返済方式の明細表が必要かどうかのメッセージ

明細表の表示

比較表の枠を描くためのサブルーチン

(注) プロッタ・プリンタを使って明細表を作成するときは、480行と500行のPRINTをLPRINTに変えて下さい。

付表

キャラクタコード

32	SP	48	0	64	@	80	P	96	`	112	p	128	
33	!	49	1	65	A	81	Q	97	a	113	q	129	
34	"	50	2	66	B	82	R	98	b	114	r	130	
35	#	51	3	67	C	83	S	99	c	115	s	131	
36	$	52	4	68	D	84	T	100	d	116	t	132	
37	%	53	5	69	E	85	U	101	e	117	u	133	
38	&	54	6	70	F	86	V	102	f	118	v	134	
39	▼	55	7	71	G	87	W	103	g	119	w	135	
40	(	56	8	72	H	88	X	104	h	120	x	136	
41	)	57	9	73	I	89	Y	105	i	121	y	137	
42	*	58	:	74	J	90	Z	106	j	122	z	138	
43	+	59	;	75	K	91	[	107	k	123	{	139	
44	,	60	<	76	L	92	¥	108	l	124	:	140	
45	−	61	=	77	M	93	]	109	m	125	}	141	
46	.	62	>	78	N	94	^	110	n	126	~	142	
47	/	63	?	79	O	95	π	111	o	127		143	

144		160		176	ｰ	192	タ	208	ミ	224		240	火
145		161	。	177	ｱ	193	チ	209	ム	225		241	水
146		162	「	178	ｲ	194	ツ	210	メ	226		242	木
147		163	」	179	ｳ	195	テ	211	モ	227		243	金
148		164	、	180	ｴ	196	ト	212	ヤ	228		244	土
149		165	・	181	ｵ	197	ナ	213	ユ	229		245	♠
150		166	ヲ	182	ｶ	198	ニ	214	ヨ	230		246	♥
151		167	ｱ	183	ｷ	199	ヌ	215	ラ	231		247	♦
152		168	ｲ	184	ｸ	200	ネ	216	リ	232		248	♣
153		169	ｳ	185	ｹ	201	ノ	217	ル	233		249	
154		170	ｴ	186	ｺ	202	ハ	218	レ	234		250	
155		171	ｵ	187	ｻ	203	ヒ	219	ロ	235		251	
156		172	ｬ	188	ｼ	204	フ	220	ワ	236		252	
157		173	ｭ	189	ｽ	205	ヘ	221	ン	237	年	253	
158		174	ｮ	190	ｾ	206	ホ	222	゛	238	月	254	
159		175	ｯ	191	ｿ	207	マ	223	゜	239	日	255	

キャラクタセット

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0				0	@	P	`	p				-	タ	ミ		火
1			!	1	A	Q	a	q			。	ア	チ	ム		水
2			"	2	B	R	b	r			「	イ	ツ	メ		木
3			#	3	C	S	c	s			」	ウ	テ	モ		金
4			$	4	D	T	d	t			、	エ	ト	ヤ		土
5			%	5	E	U	e	u			・	オ	ナ	ユ		♠
6			&	6	F	V	f	v			ヲ	カ	ニ	ヨ		♥
7			'	7	G	W	g	w			ァ	キ	ヌ	ラ		♦
8			(	8	H	X	h	x			ィ	ク	ネ	リ		♣
9			)	9	I	Y	i	y			ゥ	ケ	ノ	ル		
A			*	:	J	Z	j	z			ェ	コ	ハ	レ		
B			+	;	K	[	k	{			ォ	サ	ヒ	ロ	○	
C			,	<	L	¥	l	:			ャ	シ	フ	ワ	●	
D			-	=	M	]	m	}			ュ	ス	ヘ	ン	年	
E			.	>	N	^	n	~			ョ	セ	ホ	゛	月	÷
F			/	?	O	π	o				ッ	ソ	マ	゜	日	

└コントロールコード

コマンド一覧表

No.	コマンド	機　　能
1	LIST	プログラムを画面に表示する。
2	LLIST	プログラムをプリンタに印字する
3	SAVE	プログラムをカセットテープに記録する。
4	VERIFY	メモリー内のプログラムとカセットテープに録音されたプログラムの比較。
5	LOAD	カセットテープのプログラムをメモリにロードする。
6	RUN	プログラムを実行する。
7	CONT	中断されていたプログラムの実行を続ける。
8	NEW	変数、プログラムをクリアする。
9	DELETE	プログラムを部分的にけずる。
10	AUTO	ラインナンバーを自動発生する。
11	RENUM	ラインナンバーをつけなおす。

ステートメント一覧表

	NO.	ステートメント	機　　能
	1	REM	コメント
	2	STOP	プログラムの一時中断、CONT により続行する。
	3	END	プログラムの実行終了。
	4	LET	代入　（省略可）。
	5	PRINT 又は ?	ディスプレイに表示する。
	6	LPRINT 又は L ?	プリンターに印字する。
	7	INPUT	キー入力
	8	READ	"DATA" ステートメントのデータを読みとる。
	9	DATA	"READ" ステートメントにより読み込まれるデータを示す。
	10	RESTORE	"READ" ステートメントにより読み込む "DATA" ステートメントの場所を指定する。
	11	DIM	配列の宣言。
	12	ERASE	宣言された配列をクリアする。
	13	DEFFN	ユーザー関数を定義する。
	14	GOTO	分岐。
	15	GOSUB	サブルーチンへの分岐。
	16	RETURN	サブルーチンからの戻り。
	17	ON ～ GOTO ON ～ GOSUB	分岐する行番号を選択する。 分岐するサブルーチンの行番号を選択する。
	18	FOR ～ TO STEP ～	NEXT ステートメントとの間を指定された条件で繰り返す。 STEP は増し分を指定。1 の場合は省略可。
	19	NEXT	"FOR" ステートメントによる繰り返しの場所を指定。
	20	IF ～ THEN	条件分岐。
	21	CONSOLE	画面スクロールの範囲、クリック音、キャラクターセットを指定する。
	22	CLS	画面をクリアする。
	23	SCREEN	画面の切替え。
	24	COLOR	色の指定
	25	PATTERN	文字、スプライトのパターンを変更する。
	26	CURSOR	表示位置の指定。
	27	POSITION	座標の指定。
	28	PSET	点を打つ。
	29	PRESET	点で消す。
	30	LINE	線を描く。
	31	BLINE	線で消す。
*	32	CIRCLE	円を描く。
*	33	BCIRCLE	円で消す。
	34	PAINT	かこまれた範囲をぬりつぶす。
*	35	SPRITE	スプライトの位置、色、パターンを指定する。
*	36	MAG	スプライトの大きさを指定する。
	37	SOUND	効果音を発生させる。
	38	BEEP	BEEP 音を発生させる。
	39	HCOPY	テキスト画面の内容をプリンタにコピーする。
*	40	CALL	機械語 サブルーチンへ分岐する。
*	41	POKE	メモリーへの書き込み。
*	42	OUT	出力ポートに出力する。
*	43	VPOKE	VRAM にデータを書き込む。

＊印のステートメントは BASIC レベル 2 では使えません。

関　数

NO.	関　数	内　容
1	ABS(x)	X の絶対値を求める。
2	RND(x)	乱数を作る。
3	SIN(x)	x のサインを求める。
4	COS(x)	x のコサインを求める。
5	TAN(x)	x のタンジェントを求める。
6	ASN(x)	x のアークサインを求める。
7	ACS(x)	x のアークコサインを求める。
8	ATN(x)	x のアークタンジェントを求める。
9	LOG (x)	x の自然対数を求める。
10	LGT (x)	x の常用対数を求める。
11	LTW(x)	x の 2 を底とする対数を求める。
12	EXP(x)	e の x 乗を求める。
13	RAD(x)	度をラジアン単位に変換する。
14	DEG(x)	ラジアンを度単位に変換する。
15	PI	円周率を求める。
16	SQR(x)	x の平方根を求める。
17	INT(x)	x を越えない最大の整数を求める。
18	SGN(x)	x の正負符号を与える。
19	ASC(s)	文字列 s の最初のコードを数値で与える。
20	LEN(s)	文字列 s が何文字あるかを与える。
21	VAL(s)	文字列 s を数値に変換する。
22	CHR$ (x)	x の対応する文字や機能を与える。
23	HEX$ (x)	x の 16 進数文字列を与える。
24	INKEY$ (x)	キーボードが押されたかどうか調べる。キーボードが押されればその文字を与える。押されなければ 0 (NULL)
25	LEFT$ (s,x)	文字列 s の左から x 番目までの文字列を与える。
26	RIGHT$ (s,x)	文字列 s の右から x 番目までの文字列を与える。
27	MID$ (s,x,y)	文字列 s の左から x 番目から長さ y の文字列を与える。y は省略可です。その時は最後まで与える。
28	STR$ (x)	x をそれを表示する文字列に変換する。
29	TIME$	内部クロックの時刻を与える。
* 30	PEEK(x)	メモリー x 番地の内容を与える。
* 31	INP(x)	入力ポートの入力内容を与える。
32	FRE	ユーザー用メモリーエリアの残りバイトを与える。
33	SPC(x)	プリントステートメントで使用、スペースをあける。
34	TAB(x)	プリントステートメントで使用、表示位置の指定。
* 35	STICK(n)	ジョイスティック n の方向を示す。
* 36	STRIG (n)	ジョイスティック n のトリガボタンの状態を示す。
* 37	VPEEK(x)	VRAM x 番地の内容を与える。

エラーメッセージ

コンピュータにある命令をして、その命令がコンピュータにとって実行できない場合、「エラーメッセージ」が出ます。エラーメッセージは次の3種類の形で表示されます。

①ダイレクト命令のエラー

? エラーメッセージ error

②プログラム実行中のエラー

? エラーメッセージ error in 行番号

③ INPUT ステートメントによるキー入力データのエラー

? メッセージ

②は、行番号で示されたところに、エラーメッセージの内容のエラーがあって、実行が止まったことを表します。

メッセージ	内 容
System	BASIC インタプリタのプログラムの動作エラー。（普通、このエラーはあまり起こらない。）
N-formura too Complex	数値式が複雑すぎる。
S-formura too Complex	文字列式が複雑すぎる。
Overflow	値や演算結果が、コンピュータの扱える許容範囲を超えた。
Division by Zero	数値を0で割ったとき。
Function Parameter	関数の数値（パラメタ）が、コンピュータで扱えない。
String too long	1行に入れる文字の長さが255文字をオーバーした。
Stack overflow	◎カッコの使いすぎ ◎PAINTする図形が複雑すぎる ◎ユーザ定義関数が自分自身を呼び出している
Out of memory	メモリが足りない。変数が多すぎるとき、配列が大きいときなど。
Number of Subscripts	添字の個数が異常。
Valve of Subscript	添字の値がおかしい。
Syntax error	文法上のエラー。命令文を打ち間違えているとき。
Command Parameter	コマンドのパラメタが異常。
Line number over	AUTO又はRENUMにおいて行番号が65535をオーバーする。
Illegal line number	65535以上の行番号を付けた。
Line image too long	行番号が長くなりすぎる。（RENUMなど）
Undefined line number	RENUM、GOTO、GOSUB、IF～THEN、RESTORE,RUNのとき、行番号がない。
Type missmatch	代入する側と代入される側の型が合わない。数値、文字列が混じっている。（A＝"SEGA" など。これは数値変数に文字列を代入している。）

Out of DATA	READ文により読み込もうとしたが、DATA文のデータの個数が足りない。
RETURN without GOSUB	GOSUBを行なわないでRETURN文が実行された。
GOSUB nesting	GOSUBの入れ子が16重以上になった。
NEXT without FOR	NEXTに対応するFOR文がない。
FOR nesting	FOR～NEXTの入れ子が8重以上になった。
Statement Parameter	ステートメントの数値（パラメタ）が異常。
Can't continue	プログラムの内容変更などにより、CONTによる続行が不可能。
FOR variable name	FOR文のループ変数が数値変数以外（文字型変数又は配列）である。(FOR N$=0 TO 5 など。)
Array name	DIM文のパラメタが配列でない。
Redimensioned array	一度定義した配列を2重に定義しようとした。
Undefined array	未定義の配列をERASEしようとした。
No program	プログラムがコンピュータのメモリ中にないのにSAVEしようとした。
Memory writing	LOADのときのメモリへの書き込みエラー。
Device not ready	プリンタが接続されていない、または故障。
Undefined function	定義されていないユーザ関数を呼び出した。
Verifying	テープのプログラムとの比較エラー。
Illegal direct	ダイレクト命令の実行ができない。
Redo from start	INPUT文の入力データがおかしい（たとえば、INPUT Aのとき文字を入力したなど）。最初から入力し直しを要求する。
Extra ignored	INPUT文の入力データがおかしい。余分なデータが入力されたとき、そのデータは無視される。
Unprintable	上記以外のエラー。

初めてだってすぐわかる
SEGA(セガ)SC-3000(シリーズ)BASIC入門

昭和59年3月15日　初版発行　　定価 1200円

著　者　セガ・エンタープライゼス社

協　力　青柳　功
　　鶴田久美子

©1984 SEGA Enterprises Ltd

郵便番号　100
東京都千代田区霞が関 1-4-2
編集　電話 東京(504)6403
販売　電話 東京(504)6517
振替口座 東京 9-25976

ダイヤモンド・グラフィック社印刷・大日本製本
落丁・乱丁本はお取替えいたします　2034-344890-4405

ダイヤモンド社のパソコン学習書

パソコン自由自在 ビジネス版 (正・続)
ダイヤモンドOAソフト研究会 編
4/6判 (各)1200円

5語で動かすパソコン 基礎編
ダイヤモンドOAソフト研究会 編
4/6判 980円

パソコン・ビジネス革命10日間
ダイヤモンドOAソフト研究会 編
4/6判 1200円

40歳からのコンピューター
——プログラムづくり奮戦記
藤繩勝祐 著
B6判 980円

40歳 プログラムのわかり方・書かせ方
藤繩勝祐 著
B6判 980円

40歳からのビジネス・ポケコン
尾高繁樹 著
4/6判 950円

日本語 パソコン・ワープロ文書作法
安田賀計 著
4/6判 1000円

パソコンはふつうの人の味方です
——おもしろ使いこなし情報
クライン・ユーベルシュタイン著
4/6判 1000円

マイコン・ブレーンウェア入門
——BASICでできるシステム・ダイナミックス
小玉陽一 著
A5判 2300円

初めてだってすぐわかる SEGAパソコンSC-3000シリーズ BASIC入門
セガ・エンタープライゼス編 ダイヤモンド社

訂正表

P-129 10 CLS:SCREEN 2,2:COLOR15,1:CLS

340 IF ZS="¥" THEN GOSUB 1130

P-130 ~~1120 PRINT CHR$(15)~~ 消す

P-131 1130 SCREEN 1,1:CLS:CURSOR8,12:INPUT "イロ バンゴウ? 0~15";C

4030 IF Y=4 THEN Y=5

一部の本に印刷が不鮮明な箇所がありますので、お手元のプログラムリストを見直して下さい。

P-117 500 CURSOR SX,SY:PRINT"■"

P-122 4000 FOR I=0TO4:IF SH(I)<0 THEN 4050

P-144 650 CURSOR 5,18:PRINT"アナタノ ウマレテカラノ ニッスウ";DA+1;"ニメ デス "

SC-3000シリーズ

SC-3000H

ダイヤモンド社
定価＝1200円
2034-344890-4405

SEGA® パソコン SC-3000シリーズBASIC入門

第1章—コンピュータを動かす前に
第2章—プログラムを作らなくてもパソコンは動く
第3章—プログラム入門の入門
第4章—コンピュータらしい仕事をさせるプログラム
第5章—データ処理入門
第6章—プログラミングを便利にする命令
第7章—カセット・テープにプログラムやデータを保存する
第8章—コンピュータのおもしろさが倍増する「関数」
第9章—絵や図表を描くための命令
第10章—その他コンピュータの機能をフルに活用するための命令

ダイヤモンド社
定価＝1200円
2034-344890-4405