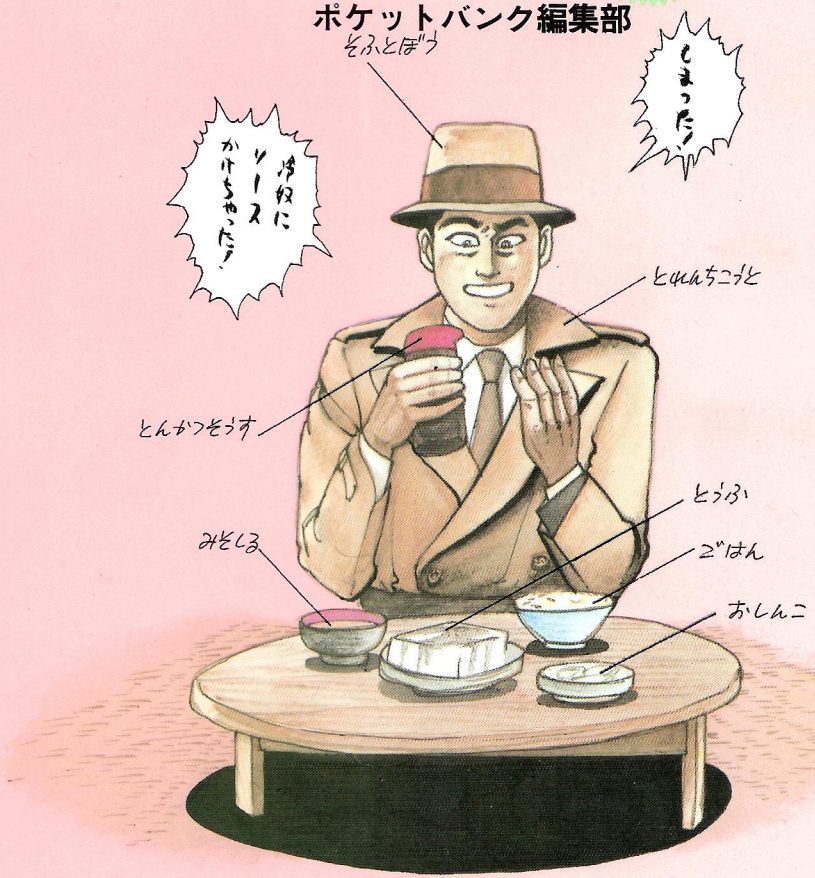
MSX POCKET BANK16

ポケットバンク

困った時の救急箱

エラー撃退ミニ事典

ポケットバンク編集部



BASICなんてもうイヤダ……
という症状には、すぐに徹底治療が必要です
そこで、トラブル退治の特効薬!

アスキー出版局

MSX POCKET BANK 16

エラー撃退ミニ事典

ポケットバンク編集部

アスキー出版局

表紙デザイン　HIT STUDIO
表紙イラストレーション　泉　昌之
本文イラストレーション　堀木康光

まえがき

全国１億１千万人のMSXファンのみなさま，こんばんはーっ，こちらポケットバンク編集部です．今月も，編集部にはみなさまからたくさんのハガキが寄せられ，一同感激の涙，涙でキーボードの上を濡らしております．

おおっと今，一枚めのハガキをＡが取り出しました．「××のプログラムを打ち込んだんだけど動かない，どーしてですか．」

これは意外だ．いきなり読者怒りのラリアートだ．これはききました．ダメージが大きい．Ａダウンだ．立て立つんだ．みんながきみをまっているのだ．あっＡが立った立ちました．ＡがんばれＡ．お，Ａ紙とペンを取り出した．凶器だ．書き殴っております．あっ紙を折り出しました．どうやら封筒固めだ……．

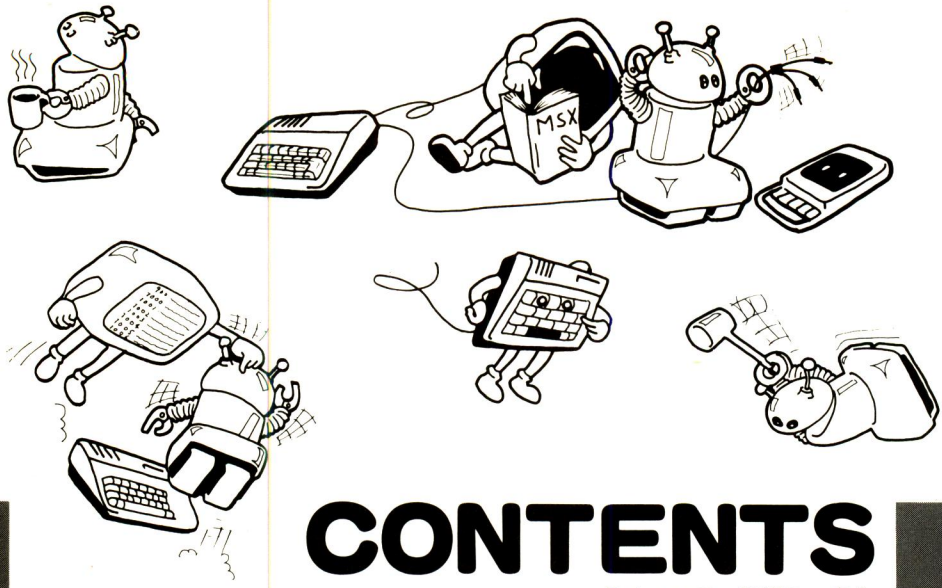
ここでスポンサーよりお知らせです．

というわけで編集部には，このような質問ハガキが多数届いている．ほおってもおけないから，毎日お返事を書いては送っていたんだけど，そのうち似たよーな質問が多いことに気が付いたんだよね．トラブるところは，結局みんないっしょのようだ．

だから，みんなが間違えそうなところをまとめて一冊の本にすれば便利だろうな，と思ったのがこの本を作るきっかけになったんだ．

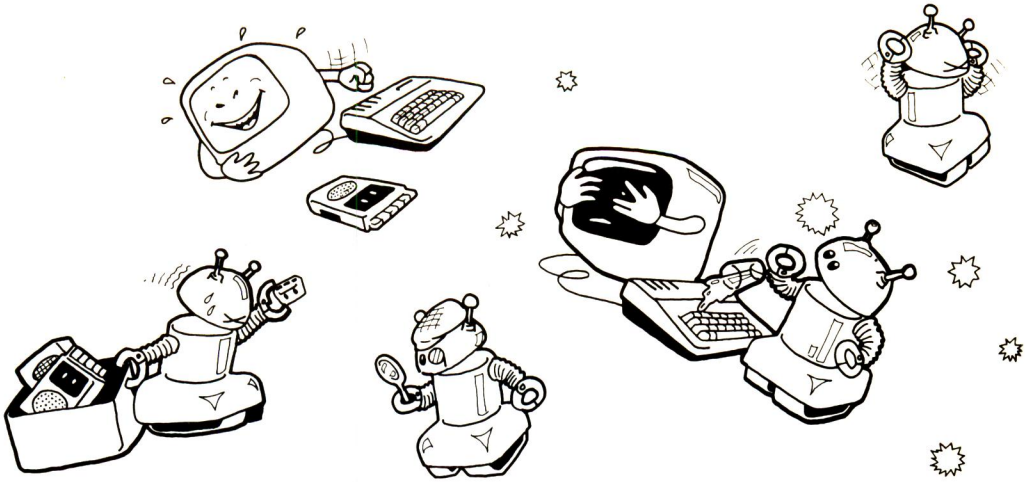
自分たち自身の経験も含め，みんなが困りそうなことへの解答が，この本にはいっぱい詰まっている．キミのギモンも，この本でズバリ解決!!

……Ａ封筒をつかんで駆け出した．ああっ，どうやらポストに入れるようであります．Ａの前にはあの赤いポストが大きな口を開けて待ち構えています．入った．入りました．Ａよくやった．額の汗が輝いています．いま再びＡは新たな戦いへと編集部に戻っていくのであります．戦えＡ，みんながきみの救いを求めているのだ……．



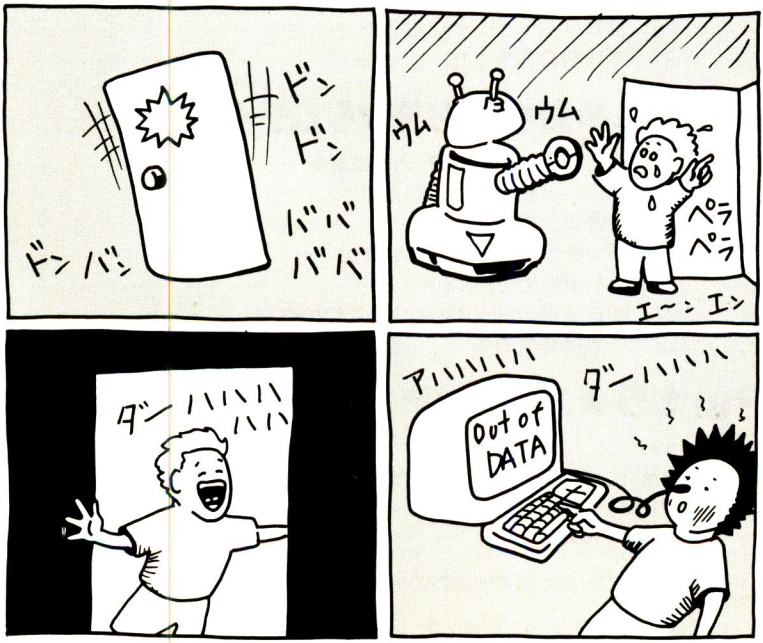
CONTENTS

目次◆エラー撃退ミニ事典



Part 1

MSXの使い方がわからない



だから，映らないわけがないと思うの

あぁ……昔はよかったな．いまじゃパソコンがデパートで買えちゃうんだもんなー．誰でも簡単に使えちゃうんだもん．あのころはさッ，ちょっとさわれるだけで町内の人気者だったんだけどな．

と，そのときである．コンコン……．

ノックの音がした．

はぁい．そこにいたのは，隣のかおる君だった．

ぐすぐすぐっすんうわわわーん……動かないよーぉん．

私は，彼の瞳をじっと見つめてこう言った．「どうしたんだい，ぼうや．」(キまったゼ，とっても)そう思いながらも，私の頭脳は彼の訪ねて来た理由を素早くキャッチしていた．

そう，これはある程度予想していた事ではあった．なんとなれば実は彼はさっき一度私の前に姿をあらわして，「かぁーちった，かっちった，MSXかっちった．いいだろー，お前なんかに見せてやんないもーん．ずぅひょひょひょー」と，言うなり飛び出していったのだ．そのときの彼のあまりの明るさに，何か不吉なものを感じたことを私は覚えている．

かわいそうに，どうやら予感があたったらしい．今の彼は，ほおがこけ，髪は白くなり廃人同様である．

こっ，これではいかん．彼のような犠牲者を，これ以上増やしてはならない．いたいけな少年少女を守らなければ．全知全霊を傾け，全財産を銀行に預けてでも，この問題と戦わなければならない．私はそう思いながら，彼にこの章を見せてあげた．すると，

「なぁんだ，うちのテレビはRF出力につながないとだめなのか．」と，彼は礼もいわずに，はっはっはっと高笑いしながら帰ってしまった．

これでよかったんだ．これで……．しかし，不幸な人は，まだまだいくらでもいるのだ．安心してはいけない．

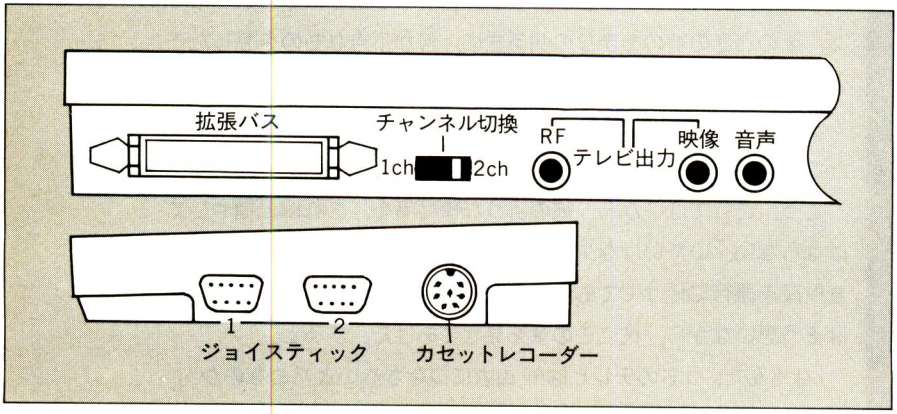
1 なーんも画面に映らない

MSXを買ってきて，テレビにつないでスイッチを入れたのに画面になーんも映らないとか，昨日はちゃんと映ったのに今日はなーんも画面に出てこないなんてことないかなあ？　こういう時はとてもアセッてしまうよね．

● つなぐ先を間違えた ●

MSXのウラ側には，画面用の出力として，たいていRF出力とビデオ出力の2種類の出力端子がついている．ほかに機種によっては，RGB用の出力がついているものもある．自分の使うテレビの種類によって，この出力を変えなければならない．これからキミが，ビデオ入力つきのテレビを使おうというのなら，MSXのビデオ出力から，テレビのビデオ端子につながなければならない．これを間違えて，RF出力からつないだりしても，なにも映らない．――えっ，そんなマヌケなことはしないって．ただ，時々テレビの電源を入れ忘れるだけだって！？

たとえば，下がPASOPIA IQ(HX-10D）のウラ側．



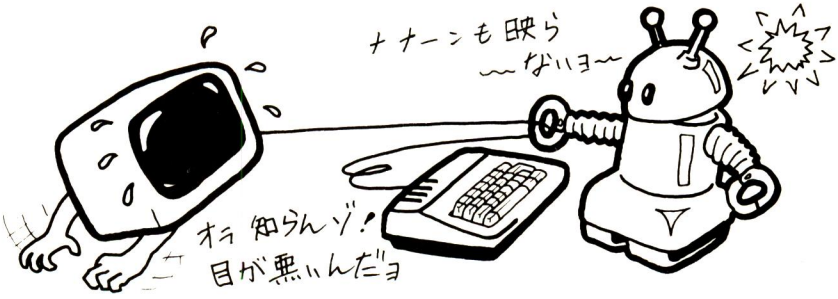
特に，ビデオ出力とは書いていないけれども，右から2番目の「映像」の端子がビデオ用．機種によっては「ビデオ」と書かれていたり，このように「映像」と書かれていたりするが，中身はおんなじ．テレビのアンテナ端子につなぐなら

RF から，ビデオ入力につなぐならビデオ出力からと，信号の形式が合っていないと画面はちゃんと映らない．また，オーディオ端子とビデオ端子を間違えていたりしたらどうしようもない．

一度セットしてしまえば問題が起きることはないが，掃除のためにコードを取りはずした時とかテレビを変えた時に，つなぎ先を間違うことがある．もし，テレビの種類が違っていたらつなぎ場所を変えたり，コネクタも変えてやらなければいけない．忘れたら取扱説明書をひっぱり出して，もう一度チェック！

● テレビのチャンネルは合っているか？ ●

端子が合っていても，画面が出てこない場合がある．たとえばRF 出力をアンテナ端子につないでいる時は，テレビのチャンネルは，1ch か2ch を使う．これが合ってなくてもダメ．MSX 本体のウラ側，または RF モジュレータに切り換えスイッチがついているから，合っているかどうか調べてみよう．



● プログラム実行中に画面が消えた？ ●

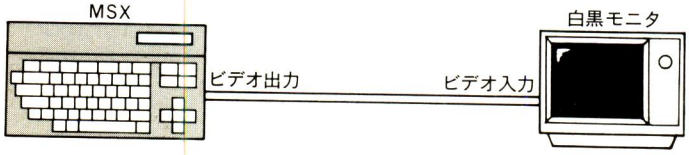
今までちゃんと映っていたのに，プログラムを RUN したら突然画面が消えてしまった．こんな場合は，プログラムに問題がある．本のリストを打ち込んだプログラムなら，SCREEN 命令や COLOR 命令の打ち込みミスによることが多い．たとえば，COLOR15，0，1となるはずが，COLOR1，0，1などと打ち間違えてたら画面にはなーんも映らない．また，自作のプログラムなら，VPOKE や BASE 関数を乱用すると，画面が映らなくなって正常の状態にもどれなくなる．そんな時は，スイッチを切って電源を入れ直せばもどるが，もちろんプログラムは消えてしまう．

2 となりのテレビは映りがいい？

— どのテレビにどんなメリットが —

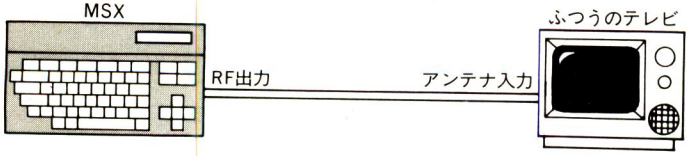
MSX用として使えるテレビにはいろいろな種類があって，それによって映り方(画面の美しさ)が大きく違う．どんな人がどんなテレビを使うのか，気になるところだ．ここでは，それを簡単に比べてみよう．

(ｉ) スーパーエコノミー・タイプ



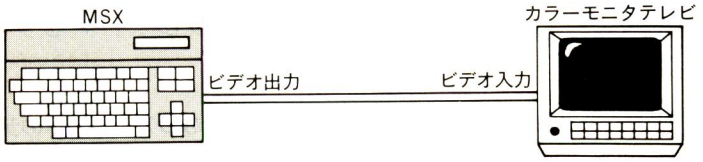
テレビもないし，カラーモニタを買うお金もない人は，パソコン用として売られているモノクロモニタをどーぞ．価格も1万円ぐらいからあるし，映りもフツーのテレビよりはずっときれいだ．BASICの入門用としてMSXを使う時は，目が疲れないという点からも，価格が安いという点からもよろしい．また，ゲームに熱中しすぎると，白黒でも，さまざまな色を区別できるようになる！？

(ⅱ) 万人のための普及タイプ



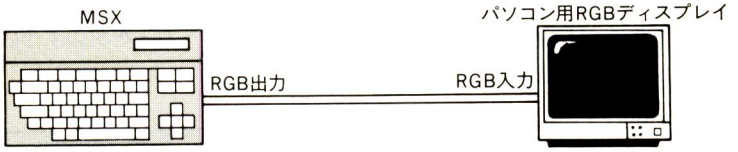
MSXユーザーの多くが，RF出力からカラーテレビのアンテナ端子につないでいる．なんたって，これがてっとり早い．RF出力の場合，MSXの画面はNHKなんかの放送と同じように，電波の形ではいってくる．利点は家庭のテレビをそのまま使えるので安くつくことだ．なお一部の機種では，MSXとTVの間に，RFコンバータというものをいれなければならない．

(iii) ポケットバンク編集部タイプ



「ビデオ入力」のあるテレビやモニタ用のテレビのうち「パソコン専用」でないものの場合，MSXのビデオ出力とつなぐと，(ii)の時よりは多少きれいに映る．ポケットバンク編集部では，もっぱらこの方法でMSXを使っている．

(iv) 画像にうるさ型向けRGBタイプ



一部のMSXには，RGB出力がついている．これで，パソコン用の高級モニタ「RGBディスプレイ」と接続できる．画面がくっきり映るという点では，(iii)より上だが，MSXでは16色が使えるのに，RGB方式では8色しか使えない．だから，一部のソフト(16色フルに使ったもの)では，困ることがあるかもしれない．

(v) なんたって最高級タイプ

サンヨーのMSX (WAVY-10) などでは，「アナログRGBディスプレイ(SONYプロフィールファインなど)」が使用できる．MSXとしては最高の組み合わせで，画面の美しさは，8色しか表現できない一般のパソコンなど及びもつかない．一度パソコンショップでその美しさを見ると，MSXのよさを再認識することができる．

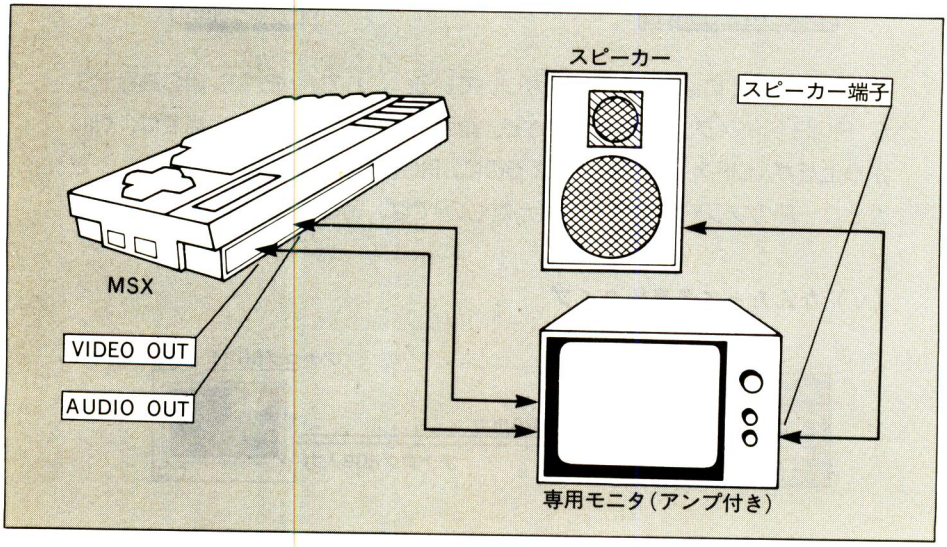
3 MSXの音が出ない

― MSXの音をグレードアップ ―

学園祭でパソコンの展示をすることになった．直前になって，ふつうのテレビでは映りが悪いからと，ビデオ入力のあるモニタテレビを買ってきたら，画面はちゃんと映るのに音が出ない．MSXの故障かと思い調べてみたが，そうではない．そもそもモニタテレビにはスピーカーがついていなかった！　こんな時，どうする？

● 音がよくなる外部スピーカー ●

一時はどうなることかと思ったけど，うまい方法を思いついた．このモニタテレビには，アンプが組み込まれていたので，愛用のスピーカーを次のように接続して音を出すことに成功したのだ．

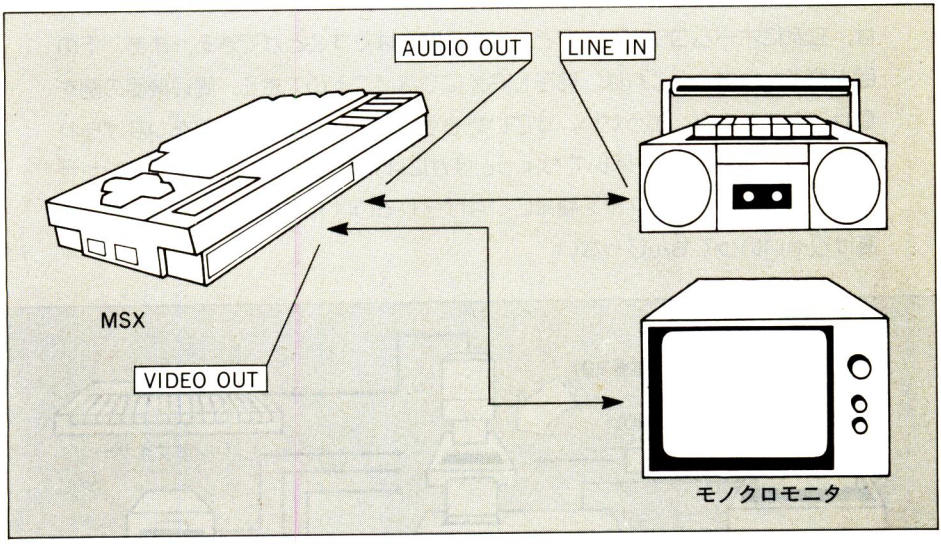


使用したスピーカは，オーディオ用として売られている超小型のもので，その片方だけを使った．安いスピーカーとはいえ，一応オーディオ用だから，音質は以前よりもぐっといい．低音もよく聞こえるようになった．これで，画面だけでなく，音までよくなったのだ．めでたし，めでたし．

● 手持ちのラジカセを利用する ●

ついでに他のケースも考えてみよう．たとえば，白黒モニタを接続する場合．この時も音は出ない．しかも，白黒モニタにはアンプなんかついていないので，アンプからそろえてやる必要がある．オーディオ用のステレオアンプとスピーカーを使えば，シブい音が出そうだが，どうも大げさすぎる．もっと手軽にラジカセを使ってみよう．

電気屋さんで「ピンプラグ－ミニプラグ」のコードを買ってきて，下の図のようにつなぐ．図のようなステレオのラジカセだと，片方のスピーカーからしか音が出ないが，そこはガマン．



MSXでは，機械のウラ側のRF端子とテレビのアンテナ端子を接続して使うというのが，もっともふつうの使い方．RFアダプターとか，テレビアダプターなどという名前のついている箱を介してテレビとつなぐ時も，これと同じこと．これらの場合は，MSXの音はテレビのスピーカーから出る．

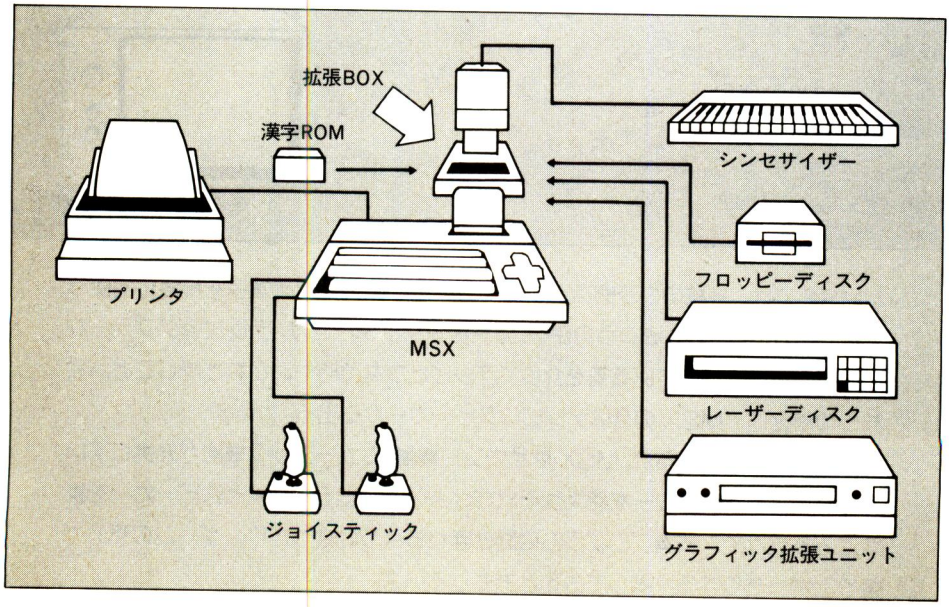
これでも用は足りるが，MSXはサウンド機能・ミュージック機能が充実しているから，よいスピーカーを使えばそれだけよい音が出る．ちょとスピーカーを換えてみるだけで，演奏プログラムに聞きほれるようになったり，ゲームの気分が盛り上がったりする．試してみるとおもしろいよ．

4 MSXはオモチャだなんて言うヤツがいたら…

「MSXなんてオモチャさ.」なんていう声を聞いたことはないかな？　そんなことを言うヤツは，MSXのほんとの使い方がわかってない！　確かにMSXには，ゲームをするのに便利な機能がたくさんついてるが，それだけじゃない！

● スロットの威力を知らないな ●

MSXはゲームしかできないと思ってる人がいれば，とんでもない話だ．MSXは，立派にホームコンピュータとしての役割をはたすことができる．まあ，下の図を見てもらおう．これは，現在MSXにつなぐことのできる，周辺機器の例を載せてみたものだ．どうだい，すごいだろう．こんなに多くの機器をつなげられるパソコンは，MSXを除いてちょっとほかには見あたらない．とくに，シンセサイザーやレーザーディスクを簡単につなげられるのはMSXだけだ．パソコンは高けりゃいいってもんじゃない．



MSX規格に合っていれば，どの周辺機器も手持ちのMSXに簡単につなぐことができる．これはスロットという共通のソケットを持っているからだ．MSXのスロットは，たいてい1個か2個だ．最近，周辺機器が出そろってきたせいもあって，スロットが1個では困るような場合も出てきた．たとえば，

〔フロッピーディスク〕+〔プリンタ〕

フロッピーディスクのインターフェイスをスロットに入れたら，プリンタ・インターフェイスを入れる場所がなくなる．もっとも，プリンタ・インターフェイス内臓のMSXなら，この心配はないが．

〔漢字ROM〕+〔ワープロソフト〕+〔プリンタ・インターフェイス〕

ワープロソフトを使うためには，当然漢字ROMが必要になるし，書いたものをプリンタに打ち出してみたくもなるはずだ．そうすると，スロットが3つ要る．

こういった場合は，別売りの拡張ボックス（たとえば，東芝HX−E601）を買い求める必要がある．これを使えば，スロットがいっきに増えるから，安心してシステムが拡張できるというわけ．

● 拡張ボックスを買う前に ●

しかし，ちょっと待った．拡張ボックスを買う前に，本当にそれが必要かどうかよく考えてみよう．よけいなものはスロットから抜けばよいのだ．例えば，ゲームをするのにプリンタ・インターフェイスやフロッピーディスク・インターフェイスは必要ない．本当に必要かどうかよく考えて，上手な買いものをしよう．

最後に，スロットに関していくつか注意．スロットにさすROMカートリッジや，各種のインターフェイスはコンピュータ本体と同じ精密な電子部品でできているので大切に扱ってあげてほしい．とりわけ，スロットの内部やカートリッジの底にある接点の部分は，よごしたりしないように気をつけること．スロットの中に小石などがはいると，即，故障のもととなる．

5 キーを押しても字が出ない

— でも，アワてなさんな —

アワッ，MSX のキーを押しても画面に字が出てこない．こんな経験は誰にでもあるだろう．でもあわてることはない．キーがこわれるなんてことは，めったにあるもんじゃない．では，どんな場合に画面に字が出なくなるか考えてみよう．

● プログラム実行中の場合 ●

まず，プログラムを実行中は，押したキーが画面に表示されることはない．これが原則．ただし，INPUT 命令や LINE INPUT 命令で入力中は，プログラム実行中でも押したキーを画面に表示することができる．この命令は，たとえばポーカーゲームで賭金を入力する時なんかに使う．

ところが，INKEY $や INPUT $では押したキーは表示されないのである．さっきの 2 つの命令に似ているのでよくカン違いするのだが，次のようなプログラムを実行してみればその違いがわかるはずだ．

```
10 CLS
20 A$=INPUT$(1):BEEP
30 IF A$<>"X" AND A$<>"x" THEN 20
40 END
```

このプログラムは，キーを押すたびにピッという音を出す．これで INPUT $では，キーを押してもなにも出ないことがわかるだろうか．なお，このプログラムは“X”キーを押すと終わる．

● これが先行入力だ ●

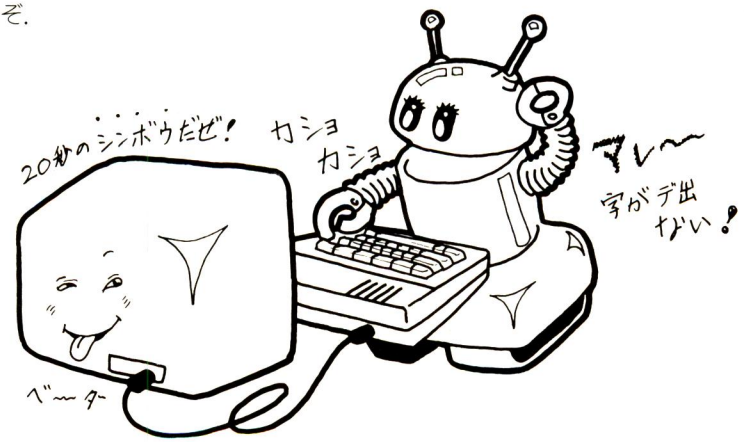
プログラムが（計算か何かを）実行中で，INPUT などのキーボードからの入力命令を実行していない時は，キーを押しても字が出ないということが，わかったと思う．おもしろいのは，この間押されたキーは，ちゃんと記憶されているということである．これを「先行入力」機能などと呼ぶ．

次のプログラムを実行してみよう．

```
10 CLS
20 FOR I=1 TO 4000
30    A=A+1
40 NEXT
50 INPUT A$
```

ここでは，20～40行で「計算」しているのだが，これには20秒ぐらいかかる．この間，画面には何も映っていない．この20秒の間にいろいろキーを押してみよう．すると，20秒後，50行の INPUT A$のところに来た時に，それまで押した文字がパッと表示されるだろう．コンピュータは，20秒間，計算をしていたのだが，同時にその間に押されたキー入力をみんな覚えているのだ．エライッ．

この「先行入力」は BASIC が正常に動いている間は，ずっと機能し続ける．たとえば，少し長いリストを表示している間に，RUN〔RETURN〕とすると，LIST 命令が終了すると同時にプログラムがスタートする．覚えておいてソンはないぞ．



TROUBLE 6

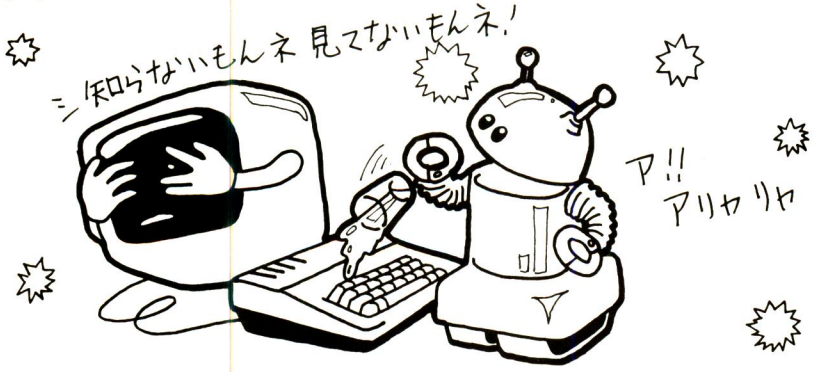
緊急事態!! ジュースをこぼしちゃった

なにー，MSXにジュースをこぼしちゃったって!? だから，あれほどジュースを飲みながらゲームするなと言ったでしょ．まったく，ひとのMSXだと思って．ほらほら，ポケットバンクなんか読んでないで，早くなんとかしてよ．事態は，一刻をあらそうんだから．

● 重点は接点 ●

ジュースをこぼしてしまったからには，はっきり言ってかなりのダメージを覚悟したほうがよいだろう．こんなことにならないように，ふだんからMSXのそばに水気のあるものを置いたり，タバコを近づけたりしないようにするべきだ．こんなことは，いやしくもコンピュータを扱う者としては，当然のことである!!

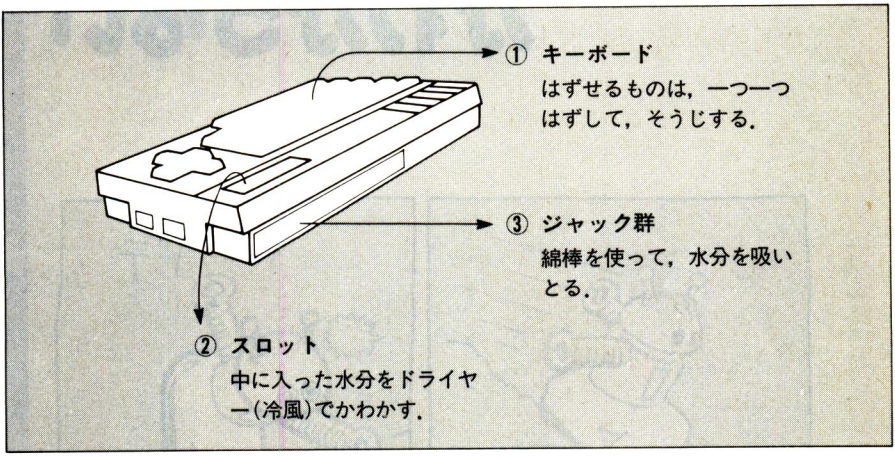
と，ここまで書いて横を見れば，となりではプログラマーのK君が，くわえタバコでプログラムのデバッグに余念がない．MSXの上には灰皿，フロッピーディスク・ドライブの上にはむきだしのディスケットとコーヒーカップが……ウーン！



MSXの中身をのぞいたことのある人は，中にゲジゲジ虫のような部品（IC）がたくさん並んでいることを知っているだろう．これらのICならば，水に対してそんなに弱くはない．ぬれても，きれいにふきとって乾かしてやればまた動くようになる．

それより，なんといっても問題なのは，電気的にくっついている「接点」なんだ．まず，①キーボード，そして②スロット，それから③本体のウラに並んでいる接続用のジャックたち！

これらの個所を重点的に掃除してやろう．かたくしぼったタオル（雑巾ではない）や，綿棒を使って可能なかぎりジュースを拭き取ってやるのだ．それが終わったら，暖かくて，直射日光のあたらないところでかわかしながら，ひたすら神様にお願いする．「神様，私は○○ゲームの3面目をやってる最中にうっかりNGを出して，あっ，いや，ジュースをこぼしてしまいました．できるかぎりの掃除はいたしました．どーか，私のMSXをお守り下さい……」



● だめなら修理 ●

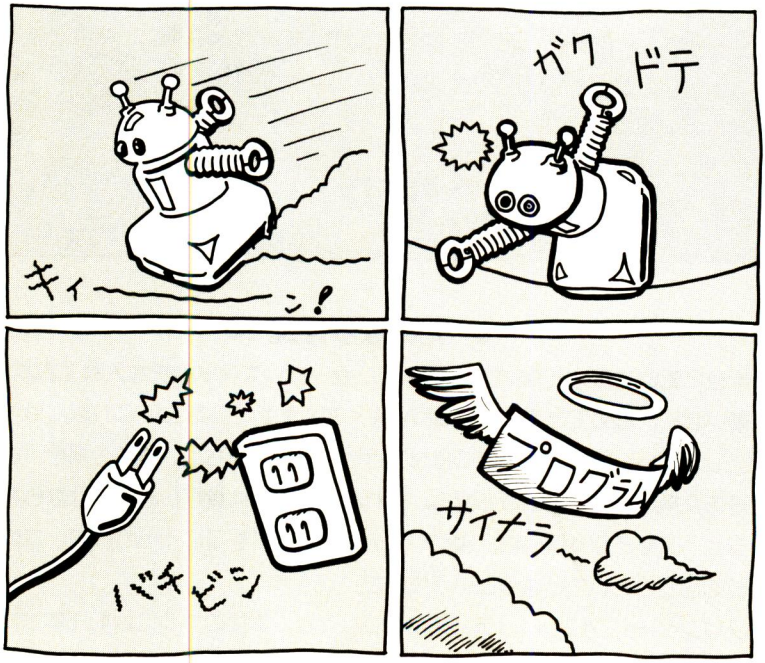
十分に乾いたら，MSXをもとのようにセットしてスイッチを入れてみよう．もし動いたら，おめでとう，キミはラッキーだ．もしダメなら，これ以上はキミにはムリだから，買った販売店に持って行って修理を頼もう．最悪の場合，キーボードの交換ということになるから，それなりの出費は覚悟しなくてはならない．

ジュースをこぼす以外に，よくやるのは，コードを足に引っかけて，本体やテレビを落っことしてしまうという失敗だ．

プリンターやフロッピーディスクをつなぐと，本体のウラには4〜5本はコードがつながることになる．ちゃんと整理して，ひっかけないようにしておかないと，たいへんなことになるよ．

Part 2

プログラムの扱い方がわからない



お母さん，ぼくのプログラム，どこへ行っちゃったんでしょうね

プログラムの打ち込みに熱中していて，学校に遅刻した……
プログラムの打ち込みに熱中していて，0点をとった……
プログラムの打ち込みに熱中していて，視力も落ちた……
プログラムの打ち込みに熱中していて，彼女にきらわれた……
プログラムの打ち込みに熱中していて，親には勘当された……

等々，プログラム入力にまつわる悲劇は数々ある．だが，そのなかで最もプログラマーの心につきささってやまない悲しみというものは，アレではないだろうか．アレって？　あれですよ．

せっかく何時間も（いや，ひょっとしたら何十時間も）かけて，苦労して打ち込んだプログラムが，ふとしたはずみで電源コードに足をひっかけたばかりにきれいさっぱりこの世から消滅してしまった時．いや，もっと悲惨なのは，ちゃんとカセットにセーブしたと思い込んで，自らの手で電源をおとしてしまった後，セーブに失敗したことに気付いた時．

いやあ何がショックかって，これほどプログラマーにとってショックなことはない．そういう事態を体験したことのある読者ならばきっとこの気持をわかっていただけるのではあるまいか．

プログラムは，各プログラマーの知恵と汗の結晶である．大切にしてやらなくちゃいけない．それには，カセットテープレコーダとのお付き合いがもっとも大事．というわけで，この章ではプログラムとテープの関係について説明することにしよう．

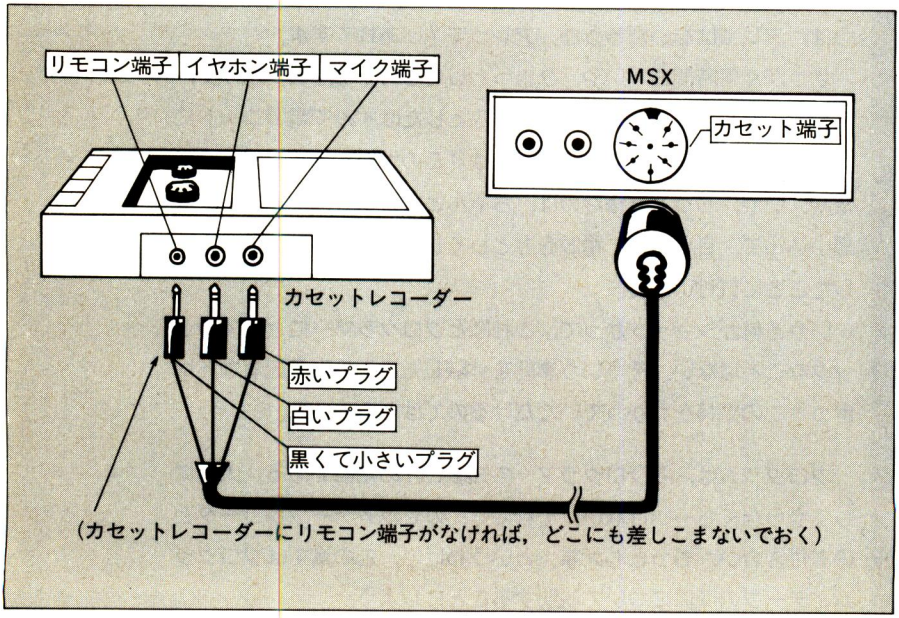
7 カセットのプログラムが読み込めない

せっかく打ち込んでセーブしておいたテープが，次の日ロードしようとしたらうまくいかない．こういう時は，とてもイライラしてしまう．

カセットがロードできない原因にはいろいろあるから，1つ1つチェックしてみよう．今まで苦労して打ち込んだプログラムがパーになるという最悪の事態がさけられるかもしれない．

● 確認1：カセットレコーダーとの接続は正しいか ●

カセット・インターフェイス・ケーブルの赤（マイク）と白（イヤホン）が，逆になっていた，というのはよくある話．また，RF アダプタ用の穴に，カセット用ケーブルをむりやり押し込んでいたという例もある．



● 確認2：ボリュームの大きさを合わせる ●

もっともよく見かけるケースだ．カセットレコーダーのボリュームの位置が悪いと，ロードエラー（Device I/O error）になったり，何も反応してくれなかったりする．ボリュームの大きさは，一般にはまん中より少し上，10段階の8くらいがいい．もちろんこの大きさは，カセットレコーダーやMSXの種類によって違ってくる．そこで，自分のテープレコーダーで，一度ちょうどいい大きさを調べておくとよい．調べるコツは，何度もロードをくり返してみて，大きすぎてエラーになる目盛と小さすぎてロードできなくなる目盛を調べて，その中間にセットする．

● 確認3：ロードの仕方を間違っていないか ●

自分のプログラムはロードできるのに，友達から借りたプログラムがロードできなかった，なんていう経験のある人は多いんじゃないかな．MSXの場合，プログラムをセーブする時はCSAVE（シーセーブ）かSAVE（セーブ）をよく使う．間違えやすいのはCSAVEで記録したものを，LOAD（ロード）で読もうとしたり，SAVEで記録したものをCLOAD（シーロード）で読もうとしたケースだ．記録方式が違ってるとロードできない．そんなおそれがあれば，ロードの形式を変えてもいちど読ませてみよう．

● 確認4：メモリは十分か？ ●

ロード中に「Out of memory」のエラーが出た場合，キミのMSXでは動かないプログラムかもしれない．同じMSXでもメモリの大きさに違いがあるから，プログラムのサイズが大きいと，メモリの小さなMSXにははいりきらない．例えばポケットバンク⑪「とにかく速いマシン語ゲーム集」の「ウミンバ」ゲームは，32K（64K）のMSXでないと動かない．8Kや16KのMSXを使ってる人はプログラムを打ち込む前に注意書きをよく読もう．8Kや16KのMSXで32K用のソフトを動かす時は，別売りの増設メモリカートリッジが必要だ．

● 確認5：ほんとにMSXのテープ？ ●

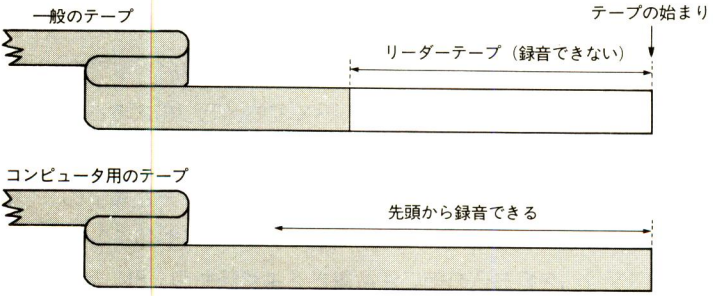
もしかしたら，そのプログラム，ほかのマシン用じゃないのかな？　例えば，FM-7とかPC-8801用のソフトテープだったりして……．他の機種のプログラムを読み込ませようとしても，記録方式が違うのだから，どだいムリ．思いあたるふしがあれば，すなおにあきらめよう．

8 カセットにうまくセーブできない

不思議なもので世の中には，プログラムをセーブするのがなぜかヘタだという人がいる．こういう人はセーブというと異常に慎重になり，あっちこっち確認し，それでも失敗して，せっかく打ち込んだプログラムをパーにする．ここでは，そんなかわいそーな人にアドバイスを．

● 注意1：セーブの前に頭出し ●

コンピュータ用カセットテープの場合，いきなりテープの頭から録音できるように，リーダーテープがついていない．ところが，音楽録音用カセットや一般のテープを使ってる場合，最初の数秒は録音できない．これを忘れるとイタイ目に会う．記録する時にプログラムの頭が欠けてしまうために，プログラムが読み込めないことになる．こうなったら，もうどうしようもない．プログラムを記録した後で，“CLOAD？”を実行する習慣や，プログラムを2度セーブする習慣をつけておくと，まさかの時でも泣かずにすむ．



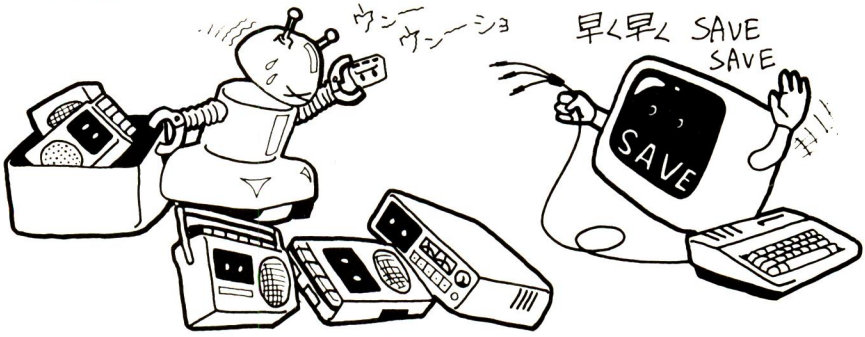
● 注意2：ピッカピカの高級テープは曲者（くせもの）だ ●

音楽専用テープなんてのは，いかにもよく記録できそうだが，表面がピカピカの高級品は，かえって音飛びして失敗することがある．もちろん，すぐにフニャフニャになってしまうような，3本で150円なりのスーパー特売品もおすすめできない．結局，一般テープかコンピュータ専用のものが一番いい．

● 注意3：ふつうのレコーダーを使う ●

自分でセーブする時には使用するカセットレコーダーとカセットテープの種類を1種類に決めてしまう．こっちのレコーダーでセーブしたテープをあっちのデッキでロードするというのは，それだけでもだいぶ条件が悪くなっていて失敗しやすくなる．また，レコーダーもオーディオ用の「高級品」は，かえってよくないようだ．

ステレオのデッキは，音質がよいから，セーブやロードがきっとうまくいくだろうと思うと，そうでもない．ステレオでやると，再生する時に右と左の音がほんの少しズレて，その結果，音が打ち消しあったりするからだ．ふつうのレコーダーにふつうのテープが一番よい．もちろん，パソコン用のデータレコーダーならもっとよい．



● 注意4：レコーダーには相性がある ●

同じようなカセットレコーダーでも，パソコンに向くものと向かないものがあることも覚えておこう．データレコーダーでないふつうのレコーダーでは，特別にパソコンのことを考えて作ってあるわけではないから，再生した時の波形の歪(ひず)み方などの関係で，どうしてもうまくセーブ，ロードできないものがある．これはもう，機械の「相性」と思ってもらうしかない．

人から借りたテープをロードしようとする時，どっちかのレコーダーのアジマスがひどくずれていると，これまたロードできない．こんな時は，音を聞くとすぐわかるから，ドライバーでアジマスをむりやり合わせてやれば，バッチリとロードできる．しかし，これはへたをするとレコーダーを壊すことになるので，「アジマスってなーに？」というような人は，決してやらないこと．

9 CSAVEとSAVE どっちでセーブしたらいい?

BASICのプログラムをセーブするのに，命令が2つあることを知っているかな? 「CSAVE」と「SAVE」の2つだ．むずかしい言葉でいうと，CSAVEは「バイナリーセーブ」でSAVEは「アスキーセーブ」なんだ．この両者の違いについて説明しよう．

● ゲームするだけの時はCSAVEが速くて便利 ●

CSAVEとSAVEの違いが一目でわかる表を作ったので，まずそれを見てほしい．

	比較ポイント	CSAVE	SAVE
1	セーブにかかる時間	短い	長い
2	マージ	できない	できる
3	セーブしたプログラムをデータとして読む	できない	できる
4	ベリファイ(CLOAD?)	できる	できない
5	ロードの形式	CLOAD	LOAD
6	オートスタート	できない	LOAD〝ファイル名〞,Rでできる

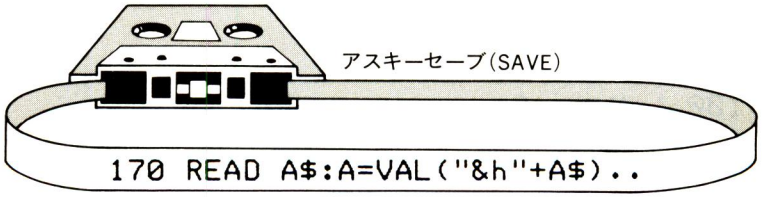
それぞれ，一長一短があることがおわかりだろう．これからもわかるように，単にセーブ，ロードするだけの時はCSAVEが速くていいし，マージなどちょっと変わったことをする時には，SAVEのほうを使わなければならないのだ．

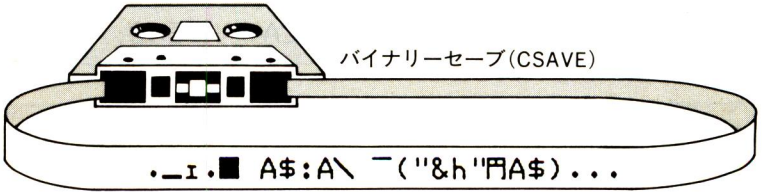
● 中間言語のヒミツ ●

MSX-BASICは，打ち込んだプログラムをそのまま文字の形でメモリに記憶するのではなく，1つ1つの命令や宣言文に番号をふった中間言語と呼ばれる一種の記号に換えて記憶している．このほうがメモリの節約になるし，実行速度をあげられるからだ．この記号化されたメモリの内容を，そのままカセットに記憶するのがCSAVEなのだ．これに対して，SAVEでは，LISTをとった時の文字の形でテープにしまう．

中間言語のままなら，プログラムの大きさは，文字のままのプログラムの半分近くですむ．そんなわけで，CSAVEはSAVEよりも，はるかに速い．

最後に，大きな声では言えないけど，アスキーセーブしたものをロードすると，一部のグラフィックキャラクタが他のものに化けてしまうというバグが現在のMSX-BASICにはある．だから，グラフィックキャラクタを使ったプログラムは，SAVEではなく，CSAVEを使わなくてはならない．気をつけよう．





10 MERGE(マージ) その傾向と対策

編集部に，「プログラムのつなぎ方がわかりません」という手紙がよくくる．これは，『トランプゲーム集』などで使われているMERGE（マージ）のことだ．マージは，キーボードからの打ち込みの手間を大幅にはぶく便利な命令だけど，使い方はチョットばかりややこしい．それぞれの本にいちおう説明してあるのだが，ここでしっかりおさらいしてみよう．

● まずはSAVE でセーブする ●

次にいくつかの短いリストが出てくるが，理屈はあとまわしにして，試しにやってみよう．習うより慣れろだ．①がもととなるプログラムで，データを②とか③のように別に作って使用する．これはポケットバンク第2巻『マイコンジュークボックス』のやり方と同じだ．実際は，②や③のデータの部分はとても長いのがふつうだ．

①と②のプログラムをマージして④のプログラムを完成させるためには，次のような手順をふむ．

準備1：まず①を打ち込み，カセットにセーブする．

```
( リスト① )

100 ' Main Program
110 SCREEN 1:WIDTH 32:CLS:READ T$
120 LOCATE 9,16-LEN(T$)¥2:PRINT T$;
130 FOR I=0 TO 13
140   READ D:SOUND I,D
150 NEXT:A$=INPUT$(1):PLAY"":END
```

```
CSAVE"MAIN" RETURN
```

準備 2：①のプログラムをNEW して消してから，②のプログラム(データだけ)を打ち込み，アスキーセーブする．

```
( リスト② )

200 ' Sound Data
210 DATA ﾍﾘｺﾌﾟﾀｧｰ
220 DATA 0,0,0,0,30,0,20,3
230 DATA 0,0,16,0,2,12
```

```
SAVE"DATA1" RETURN
```

同じ手順で下の③も“DATA 2”としてセーブしておこう．これは，あとデータを取り替える練習に使う．

```
( リスト③ )

200 ' Sound Data
210 DATA ﾌ ﾐ ｷ ﾘ
220 DATA 183,0,0,0,0,0,0,62
230 DATA 16,0,0,0,15,8
```

これで準備完了．ではいよいよ，①と②を合成して④を作ってみよう．

● いよいよマージだ ●

手順 1：まず①をロードする．①は CSAVE だからロードは CLOAD を使う．

```
CLOAD"MAIN" RETURN
```

これは CLOAD〔RETURN〕としてもいい．終わったら，一応リストをとって確認してみよう．

手順2：次に②をマージする．カセットテープを②のプログラムの先頭に合わせておいてから，

```
MERGE"DATA1" RETURN
```

を実行する．カチンカチンというリレーの音がして，しばらくするとOKが出るから，リストをとってみよう．①と②がめでたくつながって，④のようになっていれば成功だ．

```
( リスト④ )

100 ' Main Program
110 SCREEN 1:WIDTH 32:CLS:READ T$
120 LOCATE 9,16-LEN(T$)¥2:PRINT T$;
130 FOR I=0 TO 13
140   READ D:SOUND I,D
150 NEXT:A$=INPUT$(1):PLAY"":END
200 ' Sound Data
210 DATA ﾍﾘｺﾌﾟﾀｰ
220 DATA 0,0,0,0,30,0,20,3
230 DATA 0,0,16,0,2,12
```

● データだけを取り替えてみよう ●

さてマージの基本を理解してもらったところで，さらにちょっとばかり練習してみよう．今度は，メイン部はそのままで，データだけを取り替える場合だ．

手順1：同じく③をマージしてみよう．まず，④のプログラムから②の部分を削除する．プログラムの一部をけずる時は，DELETE（デリート）を使う．

```
DELETE 200-230 RETURN
```

手順2：そして，さきほどセーブした③をマージする．

```
MERGE"DATA2" RETURN
```

終わったらリストをとってみよう．データの部分だけが置き換えられたプログラムができている．

```
( リスト⑤ )

100 ' Main Program
110 SCREEN 1:WIDTH 32:CLS:READ T$
120 LOCATE 9,16-LEN(T$)¥2:PRINT T$;
130 FOR I=0 TO 13
140   READ D:SOUND I,D
150 NEXT:A$=INPUT$(1):PLAY"":END
200 ' Sound Data
210 DATA 7 ミ キ リ
220 DATA 183,0,0,0,0,0,0,62
230 DATA 16,0,0,0,15,8
```

別々に打ち込み，セーブしておいて，あとからくっつけて打ち込みの手間をはぶく．これが，マージのテクニックだ．マージはこの他にも，自分でプログラムを開発する時に，サブルーチンごとに別々に作っておいて，最後にマージして完成させる，といった使い方をすることもある．いつの日か，自分で大きなプログラムを作る時のために，心のすみに書きとめておこう．

● マージ，そのほんとのほんと ●

プログラムはふつう人間がキーボードから入力する．最初に行番号を書いて，プログラムの内容を入れ，最後にリターンキーを押すというのをくり返すわけだ．これと同じことを，カセットやディスクからするのが，マージ本来の機能なんだ．アスキーセーブされたカセットやディスクのファイルには，手で入れる時と同じように，行番号，内容，リターンというのが，ぎっしりはいっている．これをBASICはキーボードから入れたものとして処理していく．だから，マージする時は行番号の順序を考えなくてもいい．マージを「プログラムのあとに付け足す」と理解していた人，そうでないということがわかってもらえたかな？

11 このテープ，なにがはいっていたんだっけ

テープにソフトをセーブしていると，テープが20本，30本とたまっていって，だんだん整理がたいへんになる．ディスクと違って，一瞬にして中になにがはいっているか見ることができないので，ひと工夫必要だ．そこで，たまったテープの整理法を教えちゃおう．

● 「1本に1本」が理想的 ●

46分とか60分のふつうのテープを使うと，1本にたくさんのプログラムを入れてしまい，どこになにがはいっていたのかわからなくなってしまう．そんなわけで，短いテープを使って「カセット1本にソフト1本」ということにしてしまうのが，一番うまい整理法ということになる．パソコン用の10分とか15分(片面5分または7.5分)のテープがあるから，これを利用するといい．これらはテープの最初のリーダーテープがないので，録音したらプログラムの頭が欠けていたなんていうトラブルもない．

「パソコン用テープなんて，高くて使ってられない．60分のテープ1本に，みんてな押し込んでしまうのだ．という㊣のユーザーは，表を作って日頃からきちんと整理するようにしよう．最低限書くべきことは，テープカウンタの数字と内容だ．その他，ロードの方法やファイル名を記入しておくと便利だ．

テープNo.4 アクションゲーム集

カウンター	ファイル名	ロード方法	内　　容
19	GAME-A	LOAD	あーたらげーむ
85	GAME-K	BLOAD	こーたらげーむ(マシン語)
120	GAME-D	CLOAD	どーたらげーむ

● テープのプログラムを自動的にさがす方法 ●

ディスクベーシックには，「FILES（ファイルズ）」と呼ばれる命令がある．1枚のディスケットにどんなプログラムがはいっているかをいっぺんに表示させる命令だ．これと似たようなことが，テープでもできないだろうか？ なせばなる．ならぬをなすがプログラマーの道．

というわけで，プログラムの名前やファイルの形式を自動的に読んでくれる万能FILESプログラムができあがった．できたのはいいんだけど，このページにははいりきらなくなってしまったので，巻末におまけとして載せてある．詳しくはそちらを見てもらうことにして，ここではもっと簡単にファイル名を見つけ出す方法を教えよう．ただし，万能じゃないのはガマンしてもらいましょっ．

それではまず，プログラムがギシギシに詰まって，なにがなんだかわからなくなったテープを1本用意しよう．次にテープをレコーダーにかける．そして，テープを巻き戻し，プレイ状態にする．そして，

```
CLOAD"ABC" RETURN
```

として，ほっておけば画面に次から次へとファイル名が表示される．この時注意することは，“ABC”のところには，そのテープには決してはいってないファイル名を指定することだ．

```
Skip:MORITA
Skip:AKI
Skip:KAI
Skip:SAKURA
   ¦
   ¦
```

この時「SAVE（セーブ）」したものは「LOAD（ロード）」，「CSAVE（シーセーブ）」したものは「CLOAD（シーロード）」，「BSAVE（ビーセーブ）」したマシン語プログラムは「BLOAD（ビーロード）」しないと，Skip：××××の表示は出ない．つまりひとつのロード形式では同じ形式でセーブしたものしかわからない．だからCLOAD“ABC”だけでわからない場合は，LOAD“ABC”もBLOAD“ABC”もやってみる必要がある．

12 プログラムは実行前にまずSAVE

打ち込みの途中でうっかりプログラムを消してしまった，というのはよく聞く話．長いプログラムは区切りのよいところで時々セーブしておくと，こんな時でも困らない．とりわけマシン語のプログラムは，RUN させる前に必ずセーブしておくことが必要だ．

● マシン語は暴走する ●

BASIC は便利な言語だけど，スピードが遅いのが欠点だ．そのためスピードが要求されるゲームプログラムなどでは，マシン語（機械語）が使われることがある．ポケットバンク⑪『とにかく速いマシン語ゲーム集』や⑦『面白パズルブック』などの一部でもマシン語を使っている．これらのプログラムを打ち込んでみたかな？　きっと，胸のすくようなスピードに感動したことだろう．が，マシン語データを間違えた時の「暴走」の恐怖を体験した人もいるんじゃないかな？「暴走」とは，プログラムの誤り（バグ）などによって，コンピュータがメチャメチャな動作を始め，手がつけられなくなる状態のことを言う．BASIC のプログラムなら，メチャメチャな動作をしても，たいてい〔CTRL〕+〔STOP〕で止めることができるけど，マシン語が暴走したら，もはやお手あげだ．

プログラムを数日かけて打ち込み終え，「できあがり！」と思って RUN したらマシン語のデータ部分が間違っていて暴走してしまったよー，というのは泣くに泣けないよくある話．こういう場合，いくらキーを押しても止まらないから，スイッチを切るしかない．むろん，今まで打ち込んだプログラムはみんな消えてしまう．　とゆーわけで，プログラムは，実行する前に必ずセーブするクセをつけよう．こうしておけば，暴走しても大丈夫!?　暴走してもリセット（スイッチをいったん切って，5秒くらいしてから入れ直す）してから，またロードし直せばよい．　あっと，こんどは RUN しちゃダメだよ．また，暴走するだけだから．おそらく，マシン語のデータに誤りがあるはずだから，そこを中心にもとのリストと比較して誤りを見つけよう．そこって，どこなのかって？　マシン語はたいていデータ文になっている．

下のようなのがマシン語のデータだから,(スプライトのデータであることもあるが) そこを見るのだよ.

```
DATA FE,03,C0,EB,7E,23,5E,23
DATA 56,EB,57,3E,02,AE,77,23
DATA 15,20,F8,C9
```

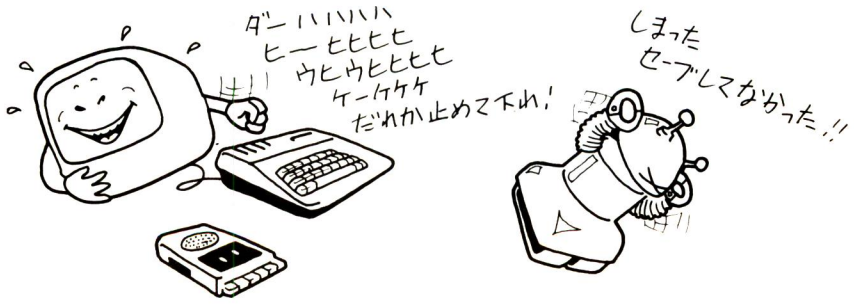
マシン語のデータは, たいてい2桁の16進数にしてあるので, 0～9, A～Fの16個の文字で表現され, それぞれコンマで区切られている. ここに他の文字がくることはない.(スプライトのデータは, マシン語のデータと間違いやすいが, そのデータを読み込むREAD文をさがすとよい.
また, データの数が不規則で最後の値がC9だとマシン語だとみてよい)

● BASICだけでもセーブせよ ●

前のとこで「BASICだけなら, 〔CTRL〕+〔STOP〕でたいてい止めることができる」と書いた. じつは, たいていというところが曲者(クセモノ)で, MSX-BASICでは, 止まらないようにすることができるのだ.

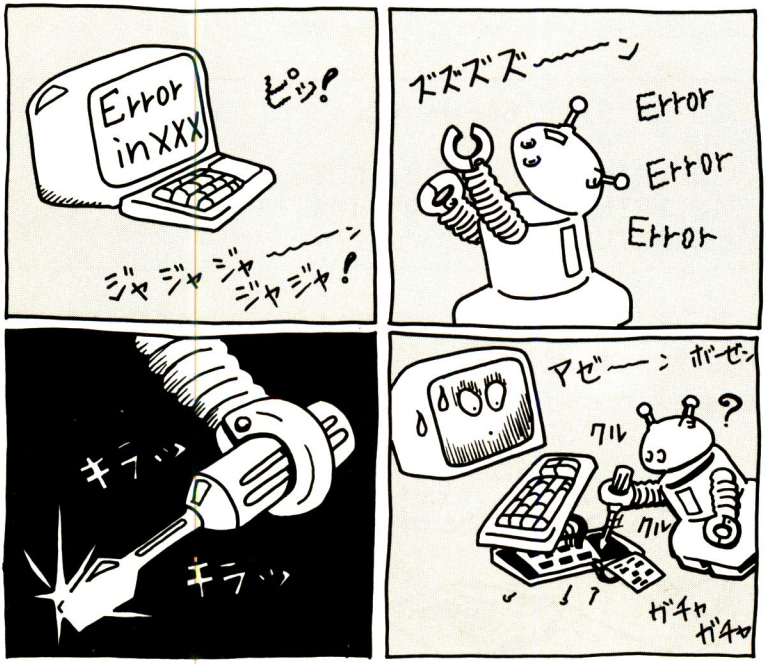
```
10 ON STOP GOSUB 40
20 STOP ON
30 PRINT"ワアーイ!!  ";:GOTO 30
40 PRINT:PRINT"トマラナイヨオ・・・"
50 RETURN
```

このプログラムは〔CTRL〕+〔STOP〕を押しても止まらない. マシン語を使ってないようだなと思って油断(ゆだん)するとこうゆうこともある. そういうわけで, 今月の標語,「プログラム, RUN(ラン)する前にまずSAVE(セーブ)」.



Part 3

エラーが出て困っている



だから，まったく動かないハズはない！

――エラーで止まってしまうのだが，いくらリストを見ても，どこが悪いのかわからない．きっと，本のリストが間違っているのだろう．おい，動かないプログラムをのせるんじゃない！ 480円返せ

編集部にはこんなハガキがチラホラくる．しかし，ちっと待ってほしい．編集部では，MSXにプログラムをロードして動くことを確認してから，プリンタにリストを打ち出し，それをこうして掲載しているのだ．

もっとも人間のやることだから，まったくミスのないプログラムだとは限らない．一部のプログラムには，実際，ちょっとした誤りも散見されている．それは「MSX通信(MSXマガジン)」でお知らせしているとおりだ．

でも，「エラーで止まってしまって，まったく動作しないプログラム」が載ることはまずない．もしエラーが発生したら，エラーの意味を本書やマニュアルを参考にして理解し，もう一度リストをチェックしてほしい．

などと，キツイことを言ったけど，誰だって打ち込んだプログラムが動かなきゃ腹も立つわいな．そこで，この章ではひとつでも多くエラーを減らす研究をしようというわけ．エラーも直す力があれば，エラーじゃないっ．

13 一番多いエラーはこれだ!!

「ポケットバンク②『マイコン・ジュークボックス』の演奏プログラムをRUNすると，Illegal function call（イリーガル・ファンクション・コール） in360と出てしまいます．360行をいくら見直しても，間違っていないのですが……」

これが，ポケットバンク編集部に寄せられたガキの中で最も多かった質問だ．

● さっそく問題の360行を見ると… ●

```
360  PLAY M$(0,P(0,I)),M$(1,P(1,I)),M$(2
,P(2,I))
```

（『ジュークボックス』58ページ）

となっていて，ここは間違っていないのに前に言ったエラーが出るというのだ．

また，『トランプゲーム集』では，

```
10340 DRAW AC$(QS)
```

（『トランプゲーム集』11ページ）

となっているところでエラーが出るというハガキがたくさんきた．この2つの質問例には，ともにマクロを使う命令が含まれている．ここで出るエラーの症状は同じなんだが，なにがエラーの原因か見当がつくかな？

● バグ探しがちょっと楽になる方法 ●

結論を言うと，質問ハガキのエラーの大半は，エラーの出た行に間違いがあるのではなく，他の行のDATA文に間違いがあった．「ああ，そうか．DATA文が違っているのか」と，改めてリストを見てギョッとした人もいるだろう．PLAYやDRAWに使うDATA文がズドドドッ……と並んでいるのだから．

DATA文の中の間違いを見つけるのは大変な仕事だ．リストを1つ1つ見直していくと，なんたって目が痛くなる．そこで，ちょっと楽をする方法を教えてあげよう．エラーが出て止まったら，その行のリストをとるのだ．

```
Illegal function call in 360
Ok
LIST 360
360  PLAY M$(0,P(0,I)),M$(1,P(1,I)),M$(2
,P(2,I))
Ok
```

そうして，行番号と「PLAY」と表示されているところに，カーソルを動かして，「PRINT」に直し，〔RETURN〕を押す．

```
PRINT      M$(0,P(0,I)),M$(1,P(1,I)),M$(2
,P(2,I))
04AF2          O3R8CACAC
O2F2.
Ok
```

表示されたのは，PLAYを実行しようとして，エラーを起こした変数の内容である．間違いはこの中にあるはずだ！この場合は，1番目の04というのがPLAYのマクロとしてはおかしい．

そこで該当するDATA文を探してみる．あった，あった．

```
11210 DATA 04AF2
```

この場合，1番目の04がほんとはO4だったのだ．

14 あやしいぞ！ まぎらわしいキャラクタたち

本のリストというのは書いてあるとおりに打ち込めば，BASICなんか知らなくても動くようにできている．にもかからわず，打ち込んだプログラムが一発で動くことなんかメッタにない．なぜか？　どうしても打ち間違いをしてしまうからだ．でも，まぎらわしいキャラクタを覚えておくと，これをぐーんと減らすことができるはず．

● まぎらわしいキャラクタたちに注意せよ ●

たいていの人が最初にやるのが「0（ゼロ）」と「O（オー）」の取り違えだ．位置も近いし(ゼロはオーの斜め上にある)，どっちも丸い形なので非常に間違いやすい．そして，リストをとってもそう簡単には気づかないから，あとで頭をかかえこむということになる．

いちいち言葉で説明するのもめんどーなので表を作った．(ホレ，右の表だ．)MSXのキーを実際に押して，自分の目で確かめよう．

書いてみて，自分でおどろいた．じつにたくさんあるものだ．「オー」と「マル」，「マイナス」と「音引き」と「横棒」なんか，ほとんど区別がつかないんじゃないかな．にもかかわらず，ちょっとコンピュータに慣れた人は間違えたりしない．どうしてだろう？

BASICに慣れてくると，この場所には，どんな文字がくるかわかってくる．たとえば，FORの次には必ず変数名がくるから，1ということはありえず，これはきっとIだろうと想像がつく．

```
FOR X=0 TO ……
```

またタイプする時，それぞれの文字を形ではなく，キーボード上の位置で覚えてしまうとタイプミスを減らすことができる．オーとゼロは，形は似ているけれど，キーの場所は違っている．どこにあるかを覚えてしまえば，間違ったりしない．じっくり研究したまえ．

		読み方	使用するキー（JIS配列）
O	O	大文字のオー	SHIFT ＋ 0 ラ
o	o	小文字のオー	0 ラ
○	O	グラフィック・キャラクタのマル	GRAPH ＋ 「 。
0	0	数字のゼロ	0 ワ ヲ
□	□	カタカナのロ	カナ ＋ ロ
I	I	大文字のアイ	SHIFT ＋ ! ニ
l	l	小文字のエル	L リ
1	1	数字のイチ	1 ヌ
｜	｜	グラフィック・キャラクタの縦棒	GRAPH ＋ ! ニ
!	!	感嘆符（かんたんふ）	SHIFT ＋ ! ヌ
-	-	マイナス	= ホ
ｰ	ｰ	音引（おんびき）	カナ ＋ ¥ ー
―	―	グラフィック・キャラクタの横棒	GRAPH ＋ = ホ
_	_	アンダースコア	SHIFT ＋ ロ
"	"	ダブル・クォート	SHIFT ＋ 2 フ
゛	゛	濁点（だくてん）	カナ ＋ " ゛
'	'	シングル・クォート	SHIFT ＋ ' ヤ ゛
`	`	バック・クォート	SHIFT ＋ ` ゛
:	:	コロン	* ケ
;	;	セミコロン	+ レ
・	・	ピリオド	> ル ゜
、	、	読点（とうてん）	カナ ＋ < ネ
・	・	中黒（なかぐろ）	カナ ＋ SHIFT ＋ ? メ
。	。	句点（くてん）	カナ ＋ > ル ゜
。	。	半濁点（はんだくてん）	カナ ＋ 「 。
+	+	プラス	SHIFT ＋ + レ
＋	＋	グラフィック・キャラクタの十字	GRAPH ＋ + レ

ポケットバンクシリーズでは，よみやすいリストをめざして，より鮮明なプリンタを研究している．
右が今回使ったもの．左が以前使って評判の悪かったもの．

15 「怪字ん」を見分けるヒケツはこうだ！

コロン（：）とセミコロン（；），ピリオド（.）とカンマ（,）の違いは，シッポがあるかないかの差なんだけど，印刷する時にシッポがつぶれて同じに見えることがある．でも，これを見破るのはそうムズカシイことではない．そのポイントをしっかり教えちゃおう．

● コロン(:)とセミコロン(;) ●

コロンというのは１つの行に2つ以上の文を書く時に使い，１つ１つの命令の区切りを示す．だから，2つ以上の命令が１行の中にある時には，次の命令の前に必ずコロンが必要だ．

```
110 SCREEN 0:COLOR 15,0,1:KEY OFF
```

セミコロンはPLAYやDRAWのマクロ，INPUT，PRINTの後ろ以外で使われることはほとんどない．

```
① PRINT"USSOOO";"HONTOOO";AC;:GOTO 100
                              ↑
② INPUT"DATA";A$              ここに注目
③ PLAY"N=A;N=B;"
④ DRAW"BM=X;,=Y;"
```

PRINT文で使われるセミコロンは，文字や変数を空白をあけずに続けて書きたい時に使う．PLAYやDRAWでは，変数のマクロを使うときに用いる．

● ピリオド(.)とカンマ(,) ●

ピリオド(.)が使われるのは次のような場合が基本．他のところにピリオドが現れたら疑ってかかろう．

```
① A=12.23:B=.3 ……変数の値
② PRINT"I am an MSX." ……文字列の1つとして
③ PLAY"O3A.BR" ……マクロ
④ DATA 12.11,2,22.2 ……小数のデータ
```

この中でやっかいなのは④の場合だ．よほど複雑な計算をするようなプログラムを別とすれば，プログラムを作るほうとしても，DATA文の中に小数のデータを入れるのをきらう．なんてたって，チェックするのがめんどーだ．そこで，もしこういうのに出くわしたら，このDATA文を読んでいるREAD(リード)文を探してみよう．もしREAD　A％とかA$とあったら，小数のデータは間違い．また，READ Aとあっても，Aがスプライトのデータや配列のカッコの中(D(A)のようなもの)に使われていれば疑ってかかるべきだ．このような場合はいずれも小数点は使えないことを覚えておこう．

● オー(O)とゼロ(0) ●

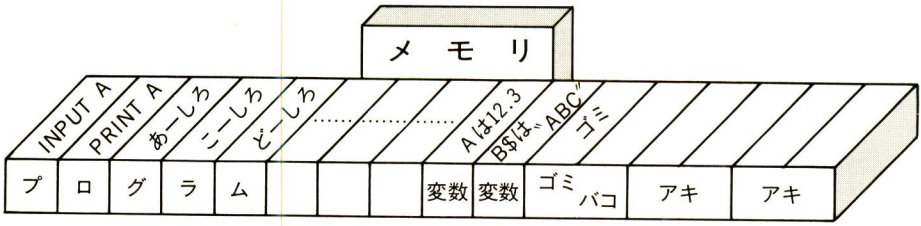
ポケットバンクの読者からよく質問を受けたのはマシン語データのOと0．マシン語のデータは，たいてい16進数で書かれている．16進数を表すのは0～9の数字とA～Fのアルファベット．だからマシン語のデータならば，この中にOがはいることはない．またプログラムを作る側から言えば，変数にOを使うこともまずない．だって，そんなことしたら自分があとで苦労するもんね．

16 メモリがたりないんですよ?

そうですか，メモリがたりないんですか．それなら，「増設メモリ」というものを買えばいいんですよ．MSX-BASICは，最大でメモリを32Kバイトにできますからね．これできっと大丈夫です．えっ，あなたのMSXはすでに32Kなの．それじゃあ，どうしようもないですね．あきらめてアスキーに手紙を出してごらんなさい．「このプログラムは，どこか間違っていないですか?」って．

● 頭のよさは,値段のよさ ●

とまあ，こんなふうに簡単にアスキーに手紙を出されてもらっても，返事を書くこっちとしても困るので，少し「メモリ」について説明しよう．



MSXは，キミが打ち込んだプログラムをちゃんと覚えていて，必要な時に（LISTと入力すると）さっと出してくれる．プログラムを実行している時でも，「今，Aという名前の変数の内容は12.3だ」といった計算の途中過程なんかもみんな覚えている．

これら「覚える」ことをやってくれるのが「メモリ」である．メモリは，多ければ多いほどよいのだが，そこはそれ，おネダンとのかねあいもあって，適当な量に決められている．○○Kバイトというのが，そのメモリの量を示している．

MSXには8Kバイト，16Kバイト，32Kバイトの3種類がある．1Kバイトは約1000文字分の「記憶の量」があることを示すから，それぞれ，約8000文字，16000文字，32000文字の記憶ができるということになる．

世の中には，64KバイトのMSXというのもある．つまり，約64000文字記憶できるわけだ．しかし，BASICを使っている限りは，BASICの都合により実際には半分しか使用できない．つまり，64Kは32Kと同じことになる．

では残りの32Kバイトは使われずに死んでいるのだろうか．イエス，BASICを使っている限りはそうである．64KのMSXはフロッピーディスクをつなぎ，MSX−DOSと呼ばれるプログラムを走らせないと，その本当の力は発揮されないのだ．

● Out of memoryは性格のはっきりしたエラーだ ●

『メモリがたりない』というエラーが出たらどうするか．あきらめるのはまだ早い．まず，プログラムの説明を見る．「このプログラムは××KバイトのMSXで動きます」という文があったら注意．きみのMSXが××Kバイト未満ならダメ．それ以上なのに，このエラーが出たならば，次のようにしてみるのが鉄則だ．カセットを使っている人はF380でCLEARするとだいたいエラーは消える．

```
CLEAR 200,&HF380 RETURN
```

をやってから，再度RUNをしてみる．これでもダメなら，打ち込みミスであろう．次のキーワードのあるところを中心に，よくリストを見比べてみよう．

```
DIM
CLEAR
GOSUB
RETURN
```

17 消えた130行

— 打ち込んだはずの行がない —

アルバイトのA氏：ねえ，このプログラム，なんかおかしいよ．

プログラマーのK氏：なにが？

A氏：ほら，RUNすると130行目がないっていうエラーで止まっちゃうんだ．

K氏：ああ．きっと130行目を打ち込み忘れたんだろう．

A氏：ところがね，こうしてLIST〔RETURN〕ってやると130行目はあるんだよ．ほら，ねっ．

K氏：……なんだコリャ？　気持悪いなあ．このMSXこわれてんじゃないの？

※ Undefined line number in240

● 消えた行の謎 ●

上にあげた話は，編集部で「行つながり」と呼んでいる現象が原因だったのだ．不慣れな人がリストを打ち込んだ時に，時々発生し，ごらんのようなミステリー性によって，長い間，その人を悩ませることになる．

編集部あてにくる質問ハガキにも，この行つながりが原因ではないか？　と考えられるものが時々ある．確認のコツさえわかれば，そうやっかいなバグでもない．そこのキミも質問ハガキを書く前にもう一度よく見てごらん．

```
100 '----- line-demo -----
110 COLOR 15,4,4:SCREEN 2:X=120
120 Y=X:DX=9:DY=DX:A=190:B=A:DA=7:DB=DA
130 E=RND(1)*8+2
140 IF X<  0 THEN DX= E
150 IF A<  0 THEN DA= E
160 IF X>255 THEN DX=-E
170 IF A>255 THEN DA=-E
180 IF Y<  1 THEN DY= E
190 IF B<  1 THEN DB= E
200 IF Y>191 THEN DY=-E
210 IF B>191 THEN DB=-E
220 X=X+DX:Y=Y+DY:A=A+DA:B=B+DB
230 LINE(A,B)-(X,Y):LINE(C,D)-(P,Q),4
240 C=A:D=B:P=X:Q=Y:GOTO 130
```

```
run

Undefined line number in 240
Ok
```

さて，上が，行つながりミステリーの証拠写真だ．じっくり見て，パズルのつもりで考えてみよう．

「130行がない!?」というエラーになってしまった．実際，LIST130〔RETURN〕とすると，すぐOkと表示され，130行がないことは確かだ．では，たんにLIST〔RETURN〕とした時に出てくる130行は，いったいなんなのか……

解説しよう．次の証拠写真2を見てもらいたい．

```
list 120
120 Y=X:DX=9:DY=DX:A=190:B=A:DA=7:DB=DA
130 E=RND(1)*8+2
Ok
```

カーソルが画面の右端までいくと，次には自動的に下の段の左端に移るのは知っているだろう．カーソルが移ったので〔RETURN〕キーを押したものと錯覚するとこういうことになるんだ．

あー，なんだ，そんなことか．と思った読者もいるだろうが，いざ実際にこんな場面に出くわすと，なにが原因かわからずにウンウンうなるはずだ．編集部のA氏でさえもこんな間違いをおかして，しばし悩んでいた．一度知ってしまえば，なんていうことのない知識も，自分で発見するとなるとえらく時間のかかるものだ．「いやー，こういうちょっとした知識が積もりに積もって，プログラマーとして一人前になれるんですよ．」とは，A氏の弁．

18 データがアウトになった!?

Out of DATA（アウト・オブ・データ）のエラーというのは，データがたりない時に出るエラーだ．と言ってしまえば，それでおしまいなんだけど，これではなんの説明にもならない．そこで，とりあえず下のリストを打ち込んでほしい．そして，これがなにをするプログラムか，ちょっと見当をつけてみよう．

● READ文がデータを読む ●

```
10 D$="ｱﾀﾞ！ ｳﾃﾄｰ"
20 FOR I=1 TO 9
30   READ X:PRINT MID$(D$,X,1);
40 NEXT
50 PRINT"  ｼﾞｬﾅｲﾓﾝﾈ"
60 DATA 7, 3, 9.2, 5, 1, 6, 8, 4
```

それでは，RUNしよう．えいっ，ピー

```
Out of DATA in 30
```

「なっなんだ?」と思った人いるかな．実はこのプログラムには，間違いが入れてあったんだ．ごめん．ほんとは，60行のデータの9と2の間は，カンマ(，)なんだよ．でも，どうしてこんなエラーが出たのかな？ いっしょに考えてみよう．

まず，60行にいくつのデータがあるか，わかるかな？ MSXは，カンマまでをひとつのデータと見るから，全部で8個だと思っているんだ．このプログラムでは，FOR～NEXTの間で，9回READ文が実行される．つまり9個のデータを読みにいくんだ．ところが9個目のデータがないことになるので，上のようなエラーが出るんだ．その証拠に“.”を“,”に直してみよう．うまくいくはずだ．

直したらRUNさせてみよう．どんなメッセージが出てきたかな？

前のは，カンマがピリオドになっていたので，データの数が少なくなっていた例だが，「Out of DATA」はDATA文ばかりでなく，FOR文に間違いがあっても出る．試しに20行を次のようにしてみよう．

```
20 FOR I=0 TO 9
```

これは，ループの回数を10回に増やすことにより，READ文も10回データを読みに行こうとするわけだけど，実際DATA文には，9個しかデータがないのでやっぱりエラーになってしまうんだよ．えっ？　そんな1を0に打ち間違えるほど，ヌケてない？　おーっと，それなら，こういうのはどうだろう．

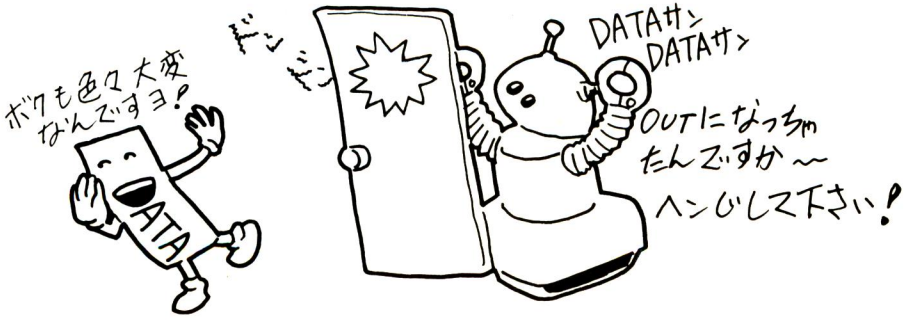
```
20 FOR I=I TO 9
```

これもやっぱり「Out of DATA」だ．なぜかというと，プログラムをRUNした瞬間，変数が全てクリアーされて0（ゼロ）になっている．つまり20行は，

```
20 FOR I=0 TO 9
```

となっているのと同じなんだ．これはひとつ前の例と同じだから，それと同じ理由で，エラーが出るってなわけだ．

Iと1は，打ち間違えキャラクタの長者番付けベスト10にはいる大物だ．間違えて当然，間違えなければもうけもの．こいつらのエラーで泣かされてきたやつを，おいらたくさん見てきたぜ．みんな，気をつけようぜ．



TROUBLE 19
絵がグチャグチャになった

ホッホッホッ，はいった．プログラムが．これで，かわゆいモンキーミミちゃんの絵が出るんだぜ．ウフフ……　それでは，RUN.

……（しばし間をおいて）

あれっ，……なんだコリャ？　絵がおかしいぞ．困ったな，どうしよう！

● 初期設定に注目 ●

とりあえず，なにが起こったか読者にも体験してもらおう．まずAのプログラムを実行して出てきた絵を覚えておく．次に，NEWしてからBのプログラムを入力し，実行する．それでもって，またNEWしてAを入れて実行してみる．するとあれれ，別の絵になった．どうしてなんだろう？

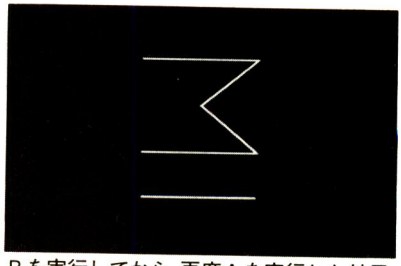
（リスト A）

```
10 SCREEN 2:COLOR 15,4,4
20 DRAW "BM100,100 C1 U10F5E
5D10"
30 DRAW "BR5 R10U5L10U5R10"
40 DRAW "BR5 F10 BL10 E10"
50 A$=INPUT$(1)
```

1回目Aを実行した結果

（リスト B）

```
10 SCREEN 2
20 FOR I=0 TO 99
30  Q=I MOD 4 : R=I¥4+1
40  DRAW"BM127,95 A=Q;S=R;E9
R9U9L9D9"
50 NEXT
60 A$=INPUT$(1)
```


Bを実行してから，再度Aを実行した結果

この絵はDRAW(ドロー)で描かれているのでとりあえず考えられるのは，DRAWマクロの初期設定が狂っていないかということだ．これは，あるプログラムを実行してから，そのあとに続けて，他のプログラムを設定した時に起こりやすい．

理由はまだわからないにしても，このように，前のプログラムの影響が次のプログラムに出てしまうことが，わかったかな．くわしい説明はあえてしない．マニュアルのDRAWのページを読んで勉強してくれたまえ．ヒントは，スケールファクタと，方向の設定は，どうなっているのかということだ．

以上のような無用のトラブルをさけるには，1つのプログラムが終わったら，リセットするクセをつけておくのがよい．

● 色バケ現象の怪 ●

絵がおかしくなったというトラブルの中には，MSXの高密度グラフィックモードで起こる「色バケ現象」がある．例によって，次のサンプルプログラムを実行してみよう．

```
10 SCREEN 2:COLOR 15,1,1:CLS
20 LINE(0,0)-(255,191),15
30 PAINT(1,0),15
40 LINE(8,0)-(247,191),12
50 A$=INPUT$(1)
```

斜めに線を引いて，上半分を白く塗る．それからまた，すこしずらして，斜めに緑色の線を引くというプログラムだ．注目すべきは，白と水色の境界線と，緑色の線がまじわったあたり．なんか変になっているだろう．ところどころ，緑色の線がにじんだようにギザギザになっている．これが色バケ現象だ．高密度グラフィックモードであるSCREEN2では，横の8つの点を1組とした枠内には，同時には2色しか表示できないのである．さっきの例では，境界線と緑色の線がまじわったあたりで，上半分の白，下半分の水色，斜めの線の緑の3色が横8ドットの中に同時にでてくるところがあって，そこがおかしくなってギザギザになったというわけ．

これは，MSXのハードウエア上の制約で，バグではないけど見苦しい．しかし，ソフトウエア上の工夫で，ほとんど目立たないようにすることはできる．今すぐどうこうできないかもしれないけど，知っておくとあとあと役に立つよ．

20 音がグシャグシャになった

ホッホッホッ，はいった．プログラムが．これで，RUN すれば宇宙刑事サイダーのテーマソングが流れるんだ．ウフフ…… それでは，RUN．

……（しばし間をおいて）

あれっ，なんだコリャ？ 変な音が出てきたぞ．困ったな，どうしよう……

（こらっ，同じリードを使うな．）

● 初期設定に注意せよ ●

音がおかしい時も，絵がおかしい時も，原因の傾向と対策は似たようなものになる．なぜなら，画面関係のトラブルメーカーである DRAW 命令と，音楽に使う PLAY 命令は，その性格がとてもよく似ているからだ．

画面に初期設定の問題があったように，音楽にもそれがある．次の例題どーぞ．

```
( リスト A )

PLAY"CDER8E8 D8R8D8C8D" RETURN
```

```
( リスト B )

PLAY"O2 S1M5000 L1 E4C" RETURN
```

前のページと同じように，まずＡを実行して音を聞き，次にＢを実行し，またＡを実行する．どうです．音が変わっちゃったでしょ．

PLAY 命令のミュージックマクロ（ＡとかＬ8とかいうやつ）の中で，プログラムを変えても，つまり，新しくロードしたり，NEW したりしても消えない設定は次のとおり．

- O（オクターブ）
- V（音の大きさ）
- M（エンベロープの早さ）
- L（長さ）
- S（エンベロープの種類）
- T（テンポ）

先ほどのAのプログラムでは，長さLも，オクターブOも，テンポTも，大きさVも，なにも設定していない．MSXのスイッチを入れた時は，長さL4，オクターブO4，テンポT120，大きさV8に設定されているので，たまたま動いたにすぎない．

実際のアプリケーションプログラムでも，このようなことが起こることがある．もっとも，このようなことの起こるプログラムは，あまりよいプログラムとは言えないが……

対策は，例によってロード前にはスイッチをいったん切ること．簡単でしょ？PLAY文に限っては，リセットするかわりに，

```
PLAY"TVOL" RETURN
```

としても同じ．リセットするばかりが初期化じゃないというのも知っとくといい．

● 音ズレのはなし ●

絵に色バケがあるなら，音に音ズレがあってもいいじゃないか（よくない！）．というわけで，音ズレの話．

```
10 PLAY"T222 L8 O4","T222 L32 O6"
20 PLAY"CE","ECECECEC"
30 GOTO 20
```

このプログラムは，20行目を永遠に繰り返す．ということは，テンポの遅いほうのパートに注目すると，ピポピポピポという音が，絶え間なく流れてるはずだ．

ところが実際は，ピポが2回鳴るごとに，少し間があいて聞こえる．まるでMSXが息つぎをしているようだ．

このように，音の長さが楽譜上の計算とずれてしまうことを「音ズレ」と読んでいる．音ズレも，BASICの一つの限界で，決め手となる対策はない．少しずつ休符を入れるなどして調節するほかない．またテンポを127にするというのも，調節に効果がある．

21 不死身の命令たち

— 覚えておこう，NEWしても解消しない命令 —

「新しくプログラムを入力したりロードする時は，リセットしてから」と何度も言ってきたけど，なぜかって言うと，NEWやCLEARをしても，前のプログラムの影響が残ってしまう命令があるからだ．この不死身の命令はマニュアルにも書いてないので，知らない人は永久に知ることはない．この項読んで覚えるべし．

● 「忘れる」命令いろいろ ●

NEWやCLEARは，新たにプログラムを打ち込み始める時や実行する時に使う命令だが，これらの命令がどう違うのかを知らなくては，お話にならない．まず，これを覚えよう．

☆ **CLEAR……**変数の内容など，BASICがたくわえておいたデータをすべて消去する．この時，飛び先行番号の情報もすべて消える．つまり，DXという変数の内容は34だったとか，次にRETURN文があったら，120行にもどる，といったことを，BASICはみんな「忘れて」しまう命令．

☆ **NEW……** CLEARでは，データは消えてしまってもプログラムは残るが，NEWではプログラムすら忘れてしまう．「ここはどこ？私はだれ？」の世界に突入してしまうおそるべき命令．世の中には，SAVEする前のプログラムを誤ってNEWしてしまった時の復活プログラムなるものまで存在する．

☆ **LOAD(CLOAD)……** カセットやディスクに保存しておいたプログラムをMSXに呼び込む命令．この場合新しいプログラムがはいると，自動的にNEWが行われる．

● 不死身のいやなやつら ●

では実際には，NEWやCLEARをしたあと，どんな命令が生き残り続けるのだろうか？　それで，どんな害があるんだろう？

★ **CLEAR のパラメータ……**「Out of memory 」や「Out of string space 」のエラーが出る.（パラメータとは，例えば，“CLEAR 300,&H DOOO”の “300” や “&H DOOO ” のこと）.

★ **PLAY 命令のL，O，S，M，T，Vの設定……**演奏速度や音階がおかしくなる.

★ **DRAW 命令のA，S，Cの設定**……絵がくずれたり，絵の色や向きがおかしくなる.

★**カーソルスイッチ……**INKEY$や PRINT の途中でカーソルが出る.
LOCATE はカーソルの位置の変更する命令だけど，もう一つの使い方がある.

LOCATE ,, 1 …… **カーソルを表示する**
LOCATE ,, 0 …… **カーソルの表示をやめる**

この設定は，プログラム実行中しか効かないが，NEW しても残っている.

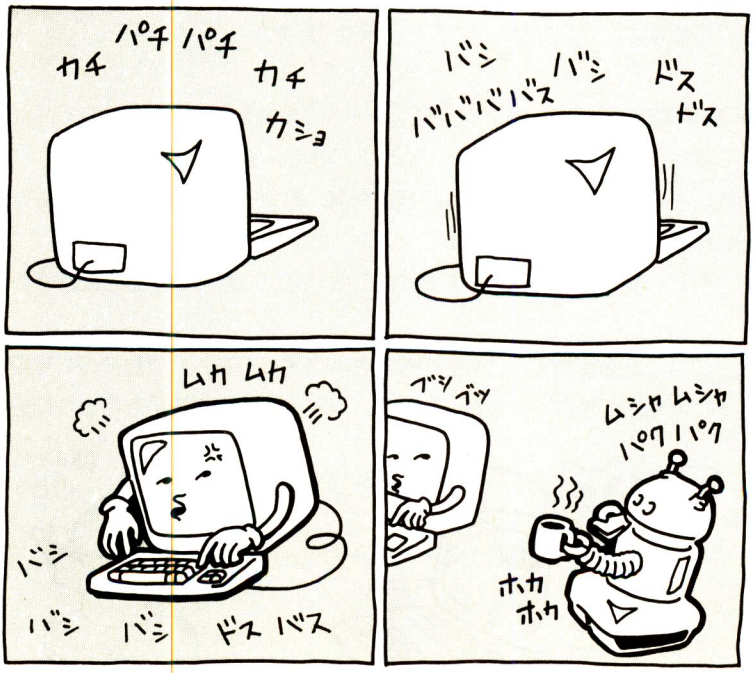
★ **SCREEN 命令……**MSX の SCREEN 命令を使うと，あきれるほどいろいろな設定ができる（最近の流行である）. つまり，画面のモード，スプライトの大きさ，キーを押した時のクリック音の有無，カセットのボーレート（セーブ，ロードの速さ），プリンタの種類だ.

これらの設定は，NEW してもすべて残っていて悪さをすることがある.

プログラムを使って，これらの設定を（スイッチを入れた直後の）初期状態にすることも，できなくはない. しかし，スイッチを入れ直したほうが何倍も早い.

Part 4

打ち込みランダムテクニック



BASICリストの入力って，こうすんのん

MSXには，ロムカートリッジという便利なものがある．スイッチを入れるだけで，ゲームやワープロが即スタートする，とても カンタン便利のスグレものだけれど，これぱかりでは進歩がない．やはり，本に載っているプログラムをキーボードから入力したり，自分でプログラムを作るようじゃないと,MSXを100％利用していると言えないものね．でも，プログラムを入れるって，いったいどうすればいいんだろう？

と言うわけで，これから，BASICの基本中の基本，プログラムのリスト入力のこつを伝授しよう．

まず，なにはともあれ，キーボードに慣れてしまうのが大切．キーボードをどのように押しても，パソコンは決して壊れたりしないから，安心していろいろやってみよう．(もちろん，うまく動かないとヒステリーをおこして強くたたけば，ぶっこわれてしまうけれど) 試しているうちに，動きが変になってしまったら，スイッチを切ってまた最初からやりなおせばいい．だから,「こんなことしたら，大切なMSXが壊れてしまうのではないだろうか」などとおくびょうがらずに，ジャンジャン触ってみるべし．

もう一つ，プログラムを入力する時は,「もっとラクな方法はないのかなぁ」と，いつも考えながらやること．人間は機械とちがってとてもものぐさ．コンピュータを作った人も，とてもものぐさ．だから，コンピュータには，ものぐさ人間が少しでもラクをするための工夫がいっぱい詰まっている．これを一つ一つ発見していくのもまた楽しい．ここでは，そういった手抜きのスペシャル技も教えてあげることにしよう．鼻唄まじりで，スラスラとリストが打ち込めるようになったら，君も一人前!?

22 正しく，リストを打ち込むために

初めてBASICのリストを打ち込んだ人にまつわる悲喜劇はいろいろある．リターンキーを押さずに画面に文字を並べただけで，リストを打ち込んだとカン違いしてしまったオジサンを見たこともあるし，リターンキーを押すのだと教えられて，R，E，T，U，R，Nと1つ1つキーを押して首をかしげた友人も知っている．そこで笑ってるキミ，ほんとに人ゴトかな？

● ちゃーんとリターンしてるかい？ ●

よくあるミスは，リストを打ち込んでいる最中に，行が画面の右端に行ってしまったのでリターンキーを押してしまい，行が切れてしまうというケース．

いずれにしても「リターンキー」を押すタイミングがカギとなっていることがわかる．BASICではリターンキーが「行」の区切りを示すから，とても重要なのである．「アッタリマエだ！」と叫んだキミ，ちっともアタリマエじゃないのだよ．世の中には，リターンキーは単にスペースと同じ意味で，単語の区切りを示すだけで，ほかになんの意味ももたない言語もたくさんあるのだ．ここで述べることは，BASIC以外にはあてはまらない，ということにも注意しよう．

なにはともあれ，ちょっと練習してみよう．まずリストを入力する時は，いったんMSX の電源を切り，再び入れ直す．こうすることによって，MSXの中が（プログラム的に）きれいに「掃除」される．そうしたら，リストを見る．

```
100 '----- line-demo 2 -----
110 DEF FNA(A,B)=B-(E-B)*(A<1)+(E+B)*(A>
MX):COLOR 15,4,4:SCREEN 2:X=120
120 Y=X:DX=9:DY=DX:A=190:B=A:DA=7:DB=DA
130 E=RND(1)*8+2
140 MX=255:DX=FNA(X,DX):DA=FNA(A,DA)
150 MX=191:DY=FNA(Y,DY):DB=FNA(B,DB)
160 X=X+DX:Y=Y+DY:A=A+DA:B=B+DB
170 LINE(A,B)-(X,Y):LINE(C,D)-(P,Q),4
180 C=A:D=B:P=X:Q=Y:GOTO 130
```

リストの ↵ のところがリターンキーを押すべきところ．130 行目なら，

1 3 0 [スペース] E =* R N D (* 1)* *

8 +* 2 RETURN

注）＊印は SHIFT を押しながら押すキー．

というふうにキーを押せばよろしい．わかっただろうか．つまり，BASIC のリストは，必ず数字（行番号）で始まるから，これをたよりにリターンキーを押していく．110行目のようにまぎらわしい個所もけっこうあるから注意．行番号は，小さい数字から，たいていは10番おきについているから，これに気をつければミスは防げる．

23 リストの確認のコツ教えます

長いプログラムリストを打ち込んだあとは，気の遠くなるような，リストの確認作業が待っている．あせって，RUNとやってしまった人も，次々と出てくるエラーメッセージに悩まされて，先へ進めない．結局，最初からリストをつき合わせるハメになる．人間が打ち込んだ以上，どっか間違えているのは当然で，その間違いを見つけ出して「つぶして」いく作業もまた当然のことである．まー，がんばってやろう．

● 画面の大きさをセットしよう ●

打ち込む時は，SCREEN1のモード(つまり，スイッチを入れた直後のモード)でやるとよいだろう．このほうが，字と字のスキマがあいて見やすくなる．

それでもって，リストを確認する時は，

```
SCREEN 0:WIDTH 40 RETURN
```

と入力する．ポケットバンクのリストはほとんど横40文字に統一してある．ということは，上のようにすると，画面のリストがポケットバンクの本のリストと同じ幅になって確認しやすい．

しかし，このモードでは多少見づらくなることがある．一つはテレビによっては画面の端(はじ)が欠けてしまうこと．もう一つは，ひらがなが読みにくくなることだ．ひらがなは，横8ドットで構成されているのだが，SCREEN 0では，このうち右端の2ドットが欠けてしまうのだ．たとえば，「い」が「し」のように見えてしまう．だから，打ち込む時には使わないほうがよい．あくまでリストの確認用と思ったほうがよいだろう．

リストをとるには，

```
LIST RETURN
```

と押す．

そうすると，リストが最初から順番に出てくる．見たいところまできたら，〔STOP〕を押すと止まる．もう一回〔STOP〕キーを押すと，再びリストが表示されていく．

ミスを見つけた時は，〔CTRL〕と〔STOP〕を同時に押すと，修正できるようになる．カーソルキー（⇦⇧⇩⇨）で直したいところにカーソルをもっていって修正し，〔RETURN〕キーを押せせばよい．

● プリンタを使おう ●

最近はプリンタもだいぶ普及しているようだ．プリンタを持っている人は，これを使わない手はないよね．LLIST〔RETURN〕とやって，リストをプリンタに打ち出してやると確認がはるかに楽になる．この時，プリンタ上のリストの幅が横80文字なら，画面の時と同じように横40文字にすれば見やすくなる．横幅の設定の変更はプリンタによって違うので，それぞれの説明書を読んでもらいたい．

ところでMSXで使えるプリンタの中には，横幅の設定が変えられないものがある．そこで，そんなプリンタを使っている人でも横40文字でリストを打ち出すことのできるプログラムを紹介しておこう．

```
100 '----- llist40 -----
110 CLEAR 1000:MAXFILES=2
120 PRINT"Ready ? then hit any"
130 A$=INPUT$(1)
140 OPEN "CAS:" FOR INPUT  AS #1
150 OPEN "LPT:" FOR OUTPUT AS #2
160 IF EOF(1) THEN CLOSE:END
170 LINE INPUT#1,A$:L=LEN(A$)/40
180 FOR I=0 TO L
190   PRINT#2,MID$(A$,I*40+1,40)
200 NEXT:GOTO 160
```

□使い方

（i）リストをとりたいプログラムをカセットにアスキーセーブしておく（SAVE "CAS：ファイル名"）．

（ii）このプログラムをロードして，RUNする．

（iii）プリンタと（i）のカセットをセットして〔RETURN〕キーを押す．

24 私はラクをしたい

— 行コピー —

ここらでひとつ，息ぬきにゲームでもしよう．下にあるのが，編集部で開発したスーパー単純数字消しゲームだ．もちろん，打ち込むのは君だよ．

● スーパー単純数字消しゲーム ●

```
100 '----- Number-Game -----
110 SCREEN 0:COLOR 15,4,4:KEY OFF
120 DEFFNS(X,Y)=VPEEK(BASE(0)+Y*40+X+1)
130 A=RND(-TIME):WIDTH 38
140 Q=9:M=M+1:T=ABS(1500-M*100):CLS
150 ON INTERVAL=T GOSUB 430
160 PRINT" Score:";P,"  Stage: ";M;
170 FOR I=1 TO 9
180 LOCATE RND(1)*29+1,RND(1)*23+1
190 IF FNS(POS(0),CSRLIN)<48 THEN BEEP:P
RINT RIGHT$(STR$(I),1); ELSE I=I-1
200 NEXT
210 INTERVAL ON:PLAY"TL32O6V"
220 A=STICK(0)
230 IF A=1 THEN DY=-1:DX= 0
240 IF A=2 THEN DY=-1:DX= 1
250 IF A=3 THEN DY= 0:DX= 1
260 IF A=4 THEN DY= 1:DX= 1
270 IF A=5 THEN DY= 1:DX= 0
280 IF A=6 THEN DY= 1:DX=-1
290 IF A=7 THEN DY= 0:DX=-1
300 IF A=8 THEN DY=-1:DX=-1
310 OX=X:OY=Y:X=X+DX:Y=Y+DY
320 IF X= 0 THEN X= 1:DX= 1
330 IF X=30 THEN X=29:DX=-1
340 IF Y= 0 THEN Y= 1:DY= 1
350 IF Y=24 THEN Y=23:DY=-1
360 B=FNS(X,Y)
370 IF B=32 OR B=111 THEN 400
380 Q=Q-1:PLAY"N=B;G"
390 P=P+10*(B-48):LOCATE 8,0:PRINT P
400 LOCATE OX,OY:PRINT" ";
410 LOCATE  X, Y:PRINT"o";
420 IF Q=0 THEN 140 ELSE 220
430 LOCATE 12,10:PRINT" Game Over "
```

□遊び方□

♧RUNすると，1〜9の数字が出てくるので，○をカーソルでコントロールして数字を消す．

♧一定時間内に数字を全部消すと一面クリアで，次の面に進める．

♧時間は面が進むにつれてどんどん短くなる．

♧カーソルをはなしても，○はとまらないから，けっこうムズカシイよ．

ずいぶんとムダの多いプログラムだ，オレならもっと短くできるって？　そーだ，そのイキだ．そうやって，他人のプログラムを研究し，自分で作り直してこそ，プログラミングの実力がついていくのだ．そんな，パワーのある人に，これから教えることは重要じゃない．単に行コピーという手抜きの方法を教えるだけなのだから．

上のリストには，なにか似たような行がたくさんある．ゲームのプログラムでは，スピードを追求するために，あえてこのような形になっていることもある．めずらしい光景ではない．

● ちょっと直して〔RETURN〕するだけ ●

たとえば，230行から300行までは似ているけど，こういう時は次のようにする．

①まず，230行を打ち込む，そして〔RETURN〕キーを押す．

```
230 IF A=1 THEN DY=-1:DX= 0 RETURN
```

②カーソルを動かして行番号を修正する．

```
240 IF A=1 THEN DY=-1:DX= 0
 ↑
 └── 修正
```

③⇨キーで違っている個所まで，カーソルを動かして修正する．〔RETURN〕もお忘れなく．

```
240 IF A=2 THEN DY=-1:DX= 1 RETURN
      ↑                  ↑
      └──── 修正 ────────┘
```

④　①～③を，繰り返す．

300行まで繰り返したら，リストを取ってみよう．見事に，230～300行がはいっている．ヤッタネ！

TROUBLE 25 じゃまなシッポは切りはなせ

今のBASICには，たいていスクリーンエディット機能がある．間違っていたら，リストをとって違ってるところだけ直して〔RETURN〕をたたくだけ！　便利になったものだ．これからお教えするのは，さらに便利に使う特殊機能のお話．

● バックスペース〔BS〕とデリート〔DEL〕 ●

その昔，パソコン（当時は，マイコンと呼ぶ人が多かった）に，字の出る画面とキーボードがついているだけでも十分にすごかった頃，BASICのプログラム入力はたいへんな作業だった．ある行が間違っているとわかった時，その間違いがたとえ1文字であったとしても，その行を最初から全部入れ直す以外に方法はなかった．とても長い行が違っていた時は，思わずため息がもれたものだ．

でも今パソコンには便利なキーがたくさんついてる．間違ったところを消してしまうのにも，〔BS〕と〔DEL〕の2種類あるのだけど，どこが違うか知っているかな？

ABC[D]EFG

このような状態で〔BS〕と〔DEL〕を押したら，それでどうなるか……ま，やってみりゃわかるから，答えは書かない．でも，このような動きを見れば，新たに打ち込んでいる時の修正は〔BS〕，いったんリストをとってからの修正は〔DEL〕が便利だというのもわかるだろう．うまく使いわければ，入力も速くなる．

● インサートはナミダもんの機能だよ ●

〔INS〕は，今話題のアイエヌエスとは，なんも関係がない．これも，言葉で説明するより，実際にやってみたほうが早い．まず，次のように入れてみる．

```
MSX Bank■
```

そこで，Bのところにカーソルをもっていって，〔INS〕を押す．カーソルの形が変わったでしょ．これは，インサート（挿入）モードになっているというしるし．そうしたら，Pocketと打ってみよう．

```
MSX Pocket Bank
```

このように，〔INS〕は，一部入力し忘れた時に使うのだ．インサートモード（押した文字が，カーソルのところにはいって，以後が右にずれる）は，リターンキーかカーソルキーを押すまで有効だ．

● シッポを消すのがイレースライン ●

さっき入力したMSX Pocket Bankの「P」のところに，カーソルを移動させて，しかる後，〔CTRL〕を押しながら〔E〕を押してみよう．

```
MSX■
```

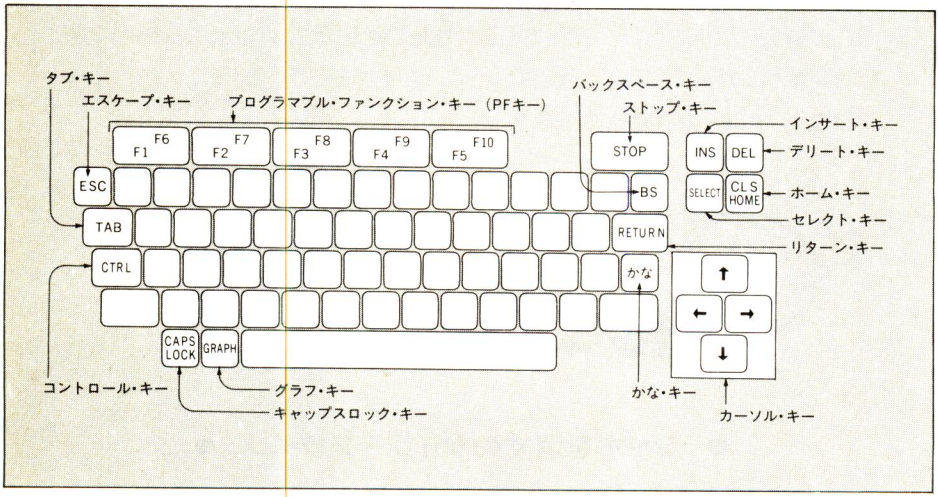
あれ，カーソルから，右側が消えてしまった．この機能をイレースラインなどと言う．今は，あまり使わないかもしれないが，カーソルを画面内で縦横無尽（じゅうおうむじん）に動かしながら，BASICを使うようになってくると，欠かせない機能となるんだ．

TROUBLE

26 これが謎のESCキー

MSXに限った話ではないが，パソコンのキーボードには，多くの特殊キーがついてる．使いこなしているかな？　キーボードを打つのに慣れてきて，「ファンクションキーに手をのばすより，ふつうに打ち込んだほうが早い(?!)」なんて言う人もいるが，ポツポツの雨だれ式タイピングの人には，特殊キーは「手抜きの網」だ．ここでそれを一挙に公開しよう．

● それぞれのキー，それぞれの機能 ●



〔STOP〕(ストップキー)……①リストをとっている時やBASICのプログラムを実行中に，このキーを押すと，一時停止をする．もう1回押せば，また動き出す．

②〔CTRL〕とともに押すと，一時停止ではなく，リストやプログラムは，中断されコマンドレベル（LISTやRUNを受けつける状態）にもどる．

なにか長いプログラムをロードして，LIST〔RETURN〕としてみよう．ナイアガラの滝のごとく，ずらずらずら〜と浜松弁でリストが出てくる．ストップ

キーを押したり，〔CTRL〕+〔STOP〕（コントロール・ストップと読むのだ）を押したりしてみよう．

〔⇧⇩⇦⇨〕(カーソルキー)……いわずもがなの，かあそるきい．「今，キーを押すと，その字はきっとここに出ますよー」という印の■（カーソル）を上下左右に自由に動かす．

〔HOME〕(ホームキー) ……このキーを押すと，カーソルは，ホームポジション（左上スミ）に一気にワープする．カーソルキーでちまちま持って行かなくてもすむ．（もっとも，ほかにはあまり用途はない．）

〔PF〕(プログラマブル・ファンクションキー) ……おどろくべきことに，〔STOP〕キー以外のどんなキーにでも変身できる上，「きまり文句」でも登録できる便利なキー．

変身はこんなふうにやる．

①〔PF1〕を「SCREEN0：LIST〔RETURN〕」にする．

```
KEY 1,"SCREEN 0:LIST"+CHR$(13) RETURN
```

②〔PF2〕を「〔RETURN〕DATA"」にする．

```
KEY 2,CHR$(13)+"DATA"+CHR$(34) RETURN
```

CHR$(13)はリターンキー，CHR$(34)は，ダブルクォーテーション（"）のことだ．ところで②はなんに使うのだろう？AUTOでたくさんの文字列データを入力する時に，これを使うと非常に便利だ．

〔ESC〕(エスケープキー)と〔SELECT〕(セレクトキー) ……この2つのキーは現在，使用されていません．というわけで，押してもなにも起こらない．しかし，これはBASICでの話であって，ゲームプログラムやMSX-DOSプログラムでは使うことがある．今のところ，関係ないかな．

27 役立たずのREM文も役に立つのだぞ

BASICには，プログラムのタイトルや，注釈などを入れる時に使うREM 文という文がある．これは，「ここに注釈がありますよ」というだけで，とくになにをするわけでもない．ところがどっこい，なにもできないだろうと思ってうかつにはぶくと，いたい目にあうぞ．

● REM文はココに気をつけよう ●

REM 文の使い方は，「REM」または「'」（シングルクォーテーション）に続いて，タイトルなどを書くだけでいい．

```
100 '***** PLAY GAME *****
500 REM ハッシャオン ノ ショリ
```

プログラムの実行中に REM の行に出くわしたら，BASIC はなにもしないで次の行に行く．だから，REM 文は多少つづりを間違えてもいいし，はぶいてもいいのかもしれない．かもしれないと言ったのは，よくない場合もあるからだ．REM はなにもしないということを聞くと，「こんなもんに用はない」と REM 文をぜーんぶはぶく人がいる．ところが REM 文をはぶくとエラーが出ることがある．

発生するエラーは，「Undefined line number（アンディファインド・ライン・ナンバー）（指定された行が見つからない）」というやつだ．下のプログラムを見てみよう．

```
100 CLS
110 REM メインルーチン
120 GOSUB 1000 'print-sub
130 GOTO 110
1000 REM サブルーチン
1010 PRINT"MSX Pocket-Bank"
1020 RETURN
```

このサンプルプログラムでは，「メインルーチン」，「Print−sub 」，「サブルーチン」という3つのREM文がある．こんな，わけのわからないREMならないほうがましだ，とか打ち込むのがメンドーくさいといって，例えば1000行を入力しなかったとしよう．どうなるだろう．

```
run RETURN
Undefined line number in 120
Ok
```

ごらんのとおりである．REM文は，きりのよいところ，たとえば，サブルーチンの入口やループのはじめなどに置かれるから，GOTO（ゴートゥー）や，GOSUB（ゴーサブ）の飛び先となることも多い．特に，プログラムをサブルーチンごとにわけて，きちんとプログラムを作る人の中には，飛び先は必ずREMにすると決めている人もいる．このプログラムの3つのREM文のうち，取り去っていいのは120行の「' Print −sub」だけだ．

結論．REM文は，できるだけはぶかないようにしよう．もし「メインルーチン」とか「サブルーチン」とか打ち込むのが面倒でも，少なくとも「REM」とだけは打ち込んでおこう．

```
100 CLS
110 REM
120 GOSUB 1000 '
130 GOTO 110
1000 REM
1010 PRINT"MSX Pocket-Bank"
1020 RETURN
```

ん，なんだって？　無口なプログラマのK君がなにか言いたいらしい．「いつも，REMのメッセージには悩（なや）んでるんだから，苦労の結晶をはぶかないでほしいよ！」だって．

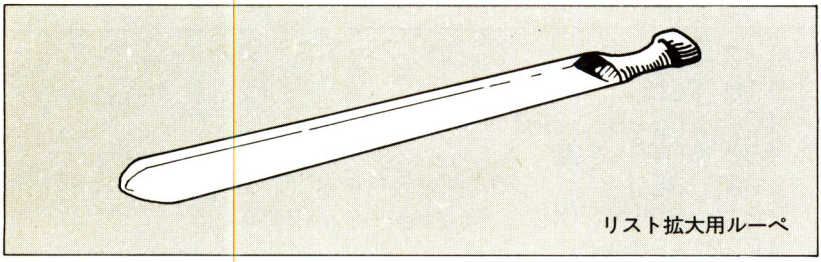
28 小道具を使おう

— より速く，より正確に —

プログラムっていうしろものは，たった１文字打ち間違えただけでも走らない．手元におもしろそうなリストがあれば，今すぐにでも打ち込んで走らせてみたいものだ．でも気をせいて，ミスをしてはなんにもならない．そこで，より速く，より正確に打ち込むための工夫を紹介してみよう．

● 定規とルーペ ●

リストを打ち込んでいて，一番困るのが，「あれ，今，どこを入れていたんだっけ？」という事態だ．行を間違えたり，同じ行を2度打ってしまうなんてのはしょっちゅうある．これに対処するには，どこまで終わったのか常にわかればいい．定規をリストの行の下に置き，リターンキーを押すたびに少しずつずらしていくといい．定規をもう一歩進めたのが下のルーペ．今から入れようとする行の上にポンと置けば，ぱっと拡大されて見える．となりの机にリストを置いても十分に見えるスグレモノなのだ．



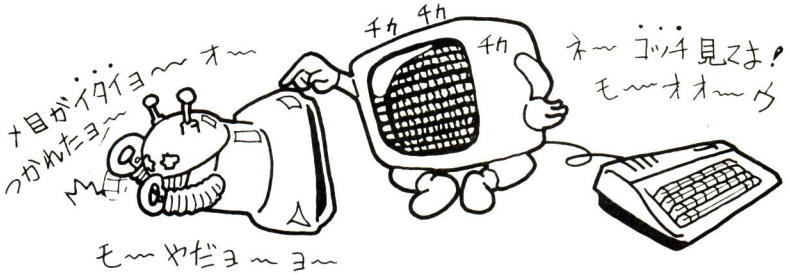
リスト拡大用ルーペ

● マーカーペンを利用しよう ●

「長いプログラムを入れる時，チェックしているうちに，本がボロボロになる．なんとかしろー．」というハガキがあった．ではなんとかしよー．コピーすれば簡単なことだ．またチェックの際はマーカーペンなんかを使うとわかりやすい．命令や変数の重要度別色わけとか，改造用変数の色わけなどをするのもおもしろい．受験勉強の要領ですな．

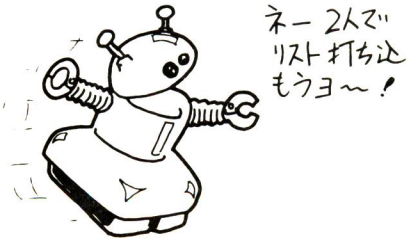
● CRTフィルタは目を護るのだ ●

長い間ディスプレイを見ていると，目がチカチカしたり肩がこったりする．これは，ディスプレイから出る紫外線のせいではないかと言われている．紫外線カットのためのCRTフィルタは，値段は張るけど，長時間ディスプレイを見る人には必需品．同じく目を護(まも)る道具として紫外線カットサングラスがあるが，あれはダサイという独断と偏見からすすめない．長いプログラムはチマチマ打ち込むよりも，長時間かかっても一気に打ち込んでしまったほうが能率がよい．だから，すわり心地のよいクッションとか，シーフードヌードルなどの目の疲れをとる健康夜食を用意しておくことも大切だ．



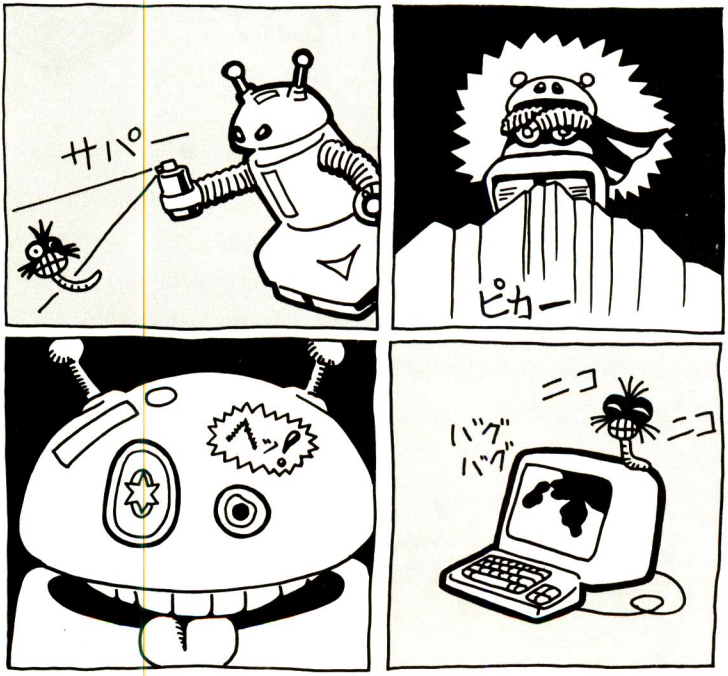
● 友だちを借りる ●

なんといっても，これは便利だ．１人より２人のほうが仕事は早く片づく．１人が打ち込んで，もう１人がその場でチェックするとか，１人が読み上げて，もう１人が打ち込んだりするといい．たとえいやな友だちでも，こういう時はニャンコの手よりはましなのだ．ついでに言うと，読み上げる時にBとDとT，MとNなんかが聞き間違えやすい．だから最初に，Tをドイツ語ふうに「テー」と読むっていうような約束をしておくといい．



Part 5

エラー対策実践教室



とにかく,やってみることですね

彼はあせっていた．仲間のバグ達は，ひとつ，またひとつと強引につぶされてゆき，いまやこのサブルーチンの中に残されているのは彼だけだった．

――どうして，どうしてなんだ．なぜオレ達ばかりが，こんな扱いを受けなくちゃならないんだ．神よ．あなたはすべての者を平等にお造りになったのではないか．ああ，もういやだ，こんな世界とはもう縁切りだ．こちらからおさらばしてやる．いや，死なばもろとも，オレはただでは消滅してやらんぞ．オレを消すというのなら，このプログラムもいっしょに破壊してやる．グシャ!!

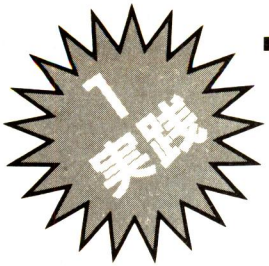
「おーい，だめだこのプログラム，暴走しちゃったぞー．」

まあ，上の話は極端な例だけど，バグ（トラブルの元となったプログラム上のミス）を力で押さえこもうとすると，往々にしてよい結果は得られない．ひとつのトラブルが解決したと思ったら，また新たな問題が生まれたりして，結局は泥沼にズブズブとのめりこんでしまうだけってことも多い．トラブルは，その原因となった部分をみつけだし，根元から絶たなきゃだめなのである．

じゃあ，トラブルの原因はどうやって見つけたらいいのかというと，なんだかんだ言っても，多くのプログラムに接してプログラムというものに慣れてもらうのが一番だ．

この章では，実際にプログラムをひとつ打ち込みながら，いままで書いてきたことをもう一度確かめてゆく．そして，いったいどんな場合にトラブルが出やすいのか，みなさんによく知ってもらいたいと思う．

「敵を知り，おのれを知れば，百のバグもあやうからず」ね！



プログラムを見切れ！

ある日，なんだか面白そうなプログラムを発見した．へえー，と思ってさっそく打ち込んではみたものの，いまイチ使い方がよくわからない．困ったなー．

……なんつってるキミ，それじゃプログラムは上達しないぞ．名プログラマーというものは，まずそのプログラムの正体をじっくりと見きわめてからキーボードに向かうものだ．そう，武道の達人が相手の力を「見切る」ようにね．

試みにプログラムをひとつ紹介しよう．さあ，キミに「見切れる」か！

● プログラムの自己提示 ●

私はプログラムである．名前だってついている．この世に生まれ出たのは，ポケットバンク編集部の中，薄暗いマシンルームの片すみでピコピコ産声(うぶごえ)をあげていたそうだ．だが，月日は流れ，バグも取れ，私もいまや一人前のプログラムに成長した．疑うひとは，88ページをごらんなさい．私の勇姿がそこにちゃあんと印刷されておる．おほん．

こうやって皆さんの前に姿を見せたのも何かの縁，「私」の入力から実行にいたるまでのプログラミング・テクニックを，私自身の口から皆さんに伝授しようではないか．プログラム自身に教えを受けるなんて，めったにない経験ですぞ．ビールの飲みかたをキリンに教わるよーなもの．これ以上いい先生いないよ．

● プログラムの内容確認 ●

そうそう．私がどんなプログラムであるのか，まだ少しも説明していないじゃないか．何をかくそう，私は，いわゆるひとつのゲームプログラムである．

どのようなゲームであるのか，基本的なコンセプトは，というと……

舞台は地面の下である．土の中には大きな岩がゴロゴロと埋もれている．その岩はとても不安定で，人が下を通ると落っこちる．落っことすと，なんと点がもらえることになっている．最終的に，より多くの点を得るのがゲームの目的だ．なお，岩が長距離を落下するほど得点も大きいから，なるべく広い空間をあけておいてから岩を落とすのが高得点をあげるコツね．

● プログラムの存在理由 ●

というわけで，私を打ち込んでくれさえすれば，（自分でこう言うのもなんだけれど），とってもおもしろいゲームが1個，手に入ってしまうのである．

えっ，いったいなぜ，『エラー撃退事典』の中に急にゲームが登場しなければならなかったのかって？　ふむ，良い質問だ．なぜ私はゲームプログラムとしてこの世に生を受けたのか？

なにしろこの章の目的は，ひとつのプログラム（私のことです）を例にとって，入力から実行にいたるまでの操作を実際に経験してもらおうというものだ．だから，ある程度リストの量があって，打ち込むのに手間がかかるようなプログラムが必要だった．別にゲームである必然性はなかったんだけれど，そのほうが打ち込みがいがあるでしょ？　まっ，そーいうわけです．

● プログラムの確認事項 ●

さて，III章やIV章では，プログラムを入力する時に確認しておきたいポイントがいくつか述べられていたはずだ．よい機会だから，そいつを「私」にもあてはめてみよう．「プログラム，打ち込み終わって泣くよりも，早めのチェックであああよかった，と一安心．（……字あまり）」

① 私は確かにMSX用に作られたプログラムである．（そりゃそおです，こうしてポケットバンクに掲載されてるんだから．でも一応は確認しておきませんとね．）

② 私は「メモリが8KのMSX」でも動作するプログラムである．（ということは，16Kでも32Kでも64Kでも当然だいじょうぶ．）

③ 私はデータ作成用の部分（リスト1），作成されたデータ（リスト2），ゲームの実行部分（リスト3）の3つに分かれており，それらを別々に取り扱わなくてはならない．（ふっふっふ，面倒な操作がいるんですよ，私って．）

④ 私は，例の「MERGE」の操作を必要とする．（なぜなら，データ作成用の部分で作り出したキャラクタ・データをゲームの部分に結合させた上で，実行されるからである．くわしくはこの先を読んでね．）

● リスト1(キャラクタフォント作成部)の仕事 ●

さて，それではプログラムの各部分について，もう少しくわしい説明を加えていくことにしよう．まず右ページのリスト1(キャラクタフォント作成部)だ．この部分は，いったい何をおこなうのか？

ポケットバンク⑨『グラフィック㊙伝』を読んだ人は知ってると思うけれど，MSXではキャラクタの形を自由に変えられる．そして，この機能を利用すれば，視覚的にも，より楽しいゲームが作れるのである．

でもキャラクタの形をDATA文に直す作業は，けっこう計算の手間がかかって面倒らしいですな．いや一人間なんて不幸なもの．その点，私のようなプログラムには計算などはお手のものだ．

そこで，まずカーソルキーなどを使って画面上にフォントを作るまでは読者の皆さんにやってもらい，それを分解し数値データに直す部分は私が引き受ける．それをおこなうのが，このリスト1の目的というわけ．

いわば，このリスト1は「プログラム自動作成プログラム」だ．(プログラムによって，別のプログラムを作り出してしまおうという試みは，ポケットバンク①巻や⑨巻でもおこなってました．み〜んな原理は同じです．)

● リスト2(データ本体)の特筆点 ●

そういう次第で，右ページのリスト2を皆さんが手で打ち込む必要は，実はまったくないのである．これはリスト1のプログラムを実行することによって私が自動的にカセットテープの上に作ってしまうものなのだ．

だけど，人が打ち込んだものだろうと，プログラムが作り出したものであろうと，生まれてきた子供に罪はない．じゃなかった，できあがったものがプログラムであることに変わりはない．それをテープからロードしてくれば，立派にプログラムとして動作する．

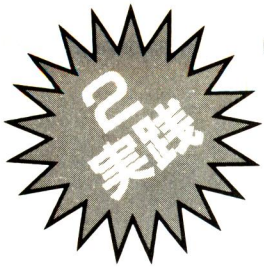
さらに良いことがある．このように「自動作成」されたプログラムは，必ずアスキーセーブされていて，もうすぐにでもマージできる体勢をととのえている．えーと，マージするためにプログラムをセーブするときには「CSAVE」を使うんだったっけ，それとも「SAVE」だったかしらん．などと迷ってポケットバンクをひっくり返す必要がないのである．

[リスト1]

```
100 SCREEN 1,,1:WIDTH 29
110 COLOR 15,4,4:CLS:PRINT"1234"
120 DEF FNS$(X)=MID$(STR$(X),2)
130 DIM D(7,7),M(3,7)
140 DV$="A:":P$=CHR$(165)+CHR$(133)
150 FOR I=4 TO 11
160   LOCATE 0,I:PRINT STRING$(8,165)
170 NEXT I
190 LOCATE X,Y+4:C=ASC(INPUT$(1))
200 IF C=13 THEN 270
210 IF C=28 THEN X=X-(X<7)
220 IF C=29 THEN X=X+(X>0)
230 IF C=30 THEN Y=Y+(Y>0)
240 IF C=31 THEN Y=Y-(Y<7)
250 IF C=32 THEN D(X,Y)=1-D(X,Y):PRINT M
ID$(P$,D(X,Y)+1,1)
260 GOTO 190
270 LOCATE 11,0:PRINT "ﾅﾝﾊﾞﾝ(0-4)?";
280 N=ASC(INPUT$(1))-48:IF N=0 THEN 350
290 IF N<1 OR N>4 THEN 280
300 LOCATE 11,0:PRINT SPC(11)
310 FOR J=0 TO 7:B=0
320   FOR I=0 TO 7:B=B*2+D(I,J):NEXT
330   VPOKE BASE(7)+384+N*8+J,B:M(N,J)=B
340 NEXT:GOTO 190
350 SCREEN 0:CLS
360 OPEN DV$+"FONT" FOR OUTPUT AS #1
370 FOR N=1 TO 4
380   PRINT #1,2000+N*10;"DATA ";
390   FOR I=0 TO 6
400     PRINT #1,FNS$(M(N,I));",";
410   NEXT:PRINT #1,FNS$(M(N,7))
420 NEXT:END
```

[リスト2]

```
2010 DATA 0,42,85,42,85,42,85,42
2020 DATA 0,0,127,0,127,0,127,0
2030 DATA 0,62,127,93,93,127,127,119
2040 DATA 0,8,28,54,99,54,28,8
```



賢いプログラム入力法

プログラムがどんなものかわかったら，入力にとりかかろう．

● 実行可能プログラム完成までの手順 ●

前のページで書いたように，プログラムの構成は多少複雑だ（これは，話をややこしくするため，わざとそうしているんです）．構成が複雑な分だけプログラムを作成するのもやっかいで，実行までには，次のような手順を踏まなくてはならない．

＊まず，空のカセットテープを4本用意する．あとあと取り違えのないように，各テープには鉛筆で1〜4の番号をふっておくといい．

＊そうしたら，まず77ページのリスト1を打ち込もう．これは，ゲームの中で使うキャラクタの形（カッコよくいえば「キャラクタ・フォント」）を作成する部分だ．

打ち込み終わったら，そのプログラムはテープ1にセーブしておく．

＊次に，いま入力したフォント作成プログラム（リスト1）を実行する．

プログラムを実行することによって，DATA文だけからなるプログラム（リスト2）が作られ，カセット（テープ2）にセーブされる．

＊その次，88ページのリスト3を入力する．これは，ゲームを実行する本体の部分だ．打ち込んだら，テープ3にセーブしておく．

＊最後に，リスト3とリスト2を「MERGE」命令を使って結合する．これでようやく最終的に実行可能なプログラムができあがる．できたプログラムをテープ4にセーブしておけば，次からはこのテープ4さえあればゲームが実行できる．

● リスト1の入力 ●

じゃあ，リスト1を入力してみよう．

1110～1140行のあたりでは，例の「行コピーのテクニック（62ページ）」が使えますな．ただしこのテクニックを使う場合，上の行と下の行で，どこに変更があるのかよく確かめて，間違いないように気をつけよう．「X」と「Y」なんて，文字の形が似ているから，画面と紙面を見くらべて，特によおーくチェックしなくてはだめだよ．

1180行以降では，「M」という文字と「N」という文字が，見た目も発音も似ているので，間違えないように．

● リスト2の作成 ●

リスト1の入力が終わったら，念のためテープ1にセーブ（何度もいうけど，プログラムは実行の前にまずセーブ）し，そしてRUN．

するとまず，タテ8×ヨコ8の大きさで点が四角く並び（これは1個のキャラクタを8倍に拡大した図である）その左上にカーソルが現れる．

カーソルキーを使って，そのカーソルを適当な位置にもっていき，スペースバーを押すと，点が黒丸に変わる．同じ位置でもう一度スペースバーを押すとまたもとの小さな点に変わる．

これを繰り返して黒丸を並べ，好きな形を作ろう．もうこれでOKだと思ったら，〔RETURN〕キーを押す．

すると，

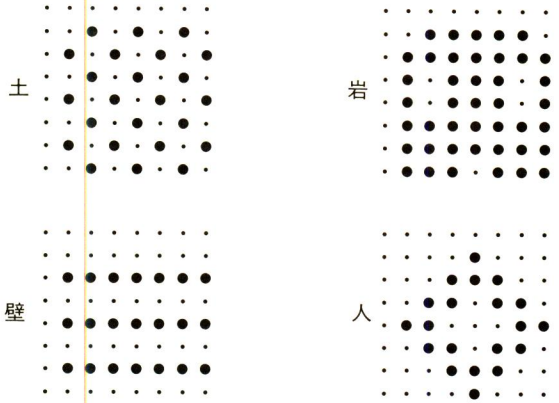
```
ナンバン ？
```

と質問するから，ここで，1～4のキーを押すと，作成した形がその番号に登録される．そして，画面の上の方に表示してある「1」～「4」の文字が，作った形に姿を変えるはずだ．

● 作成するキャラクタ・フォントの例 ●

今回ゲームで使用するキャラクタは，土，岩，人，壁の４種類だ．これらのキャラクタをどのような形に設定するかは自由である．1例として，下図のようなパターンを考えてみたので，フォントを作るときの参考にしてほしい．

またそれぞれのキャラクタは，土−1，岩−2，壁−3，人−4，の番号に登録するようになっている．



● データのセーブとその確認 ●

〔RETURN〕キーを押した時出てくる「ナンバン?」の質問に，1〜4でなく0と答えると，作成し終わった4つのキャラクタデータをカセットにセーブし始める．「0」のキーを押したとたんに，カセットの用意ができていようがいまいが，もう容赦なくセーブを始めちゃうのである．

だから順番としては，まず空テープ（テープ2）をレコーダーにセットしてレコーダーの録音ボタンを押し，さあセーブをはじめるぞっ，と心の準備をととのえてから，おもむろに「0」を押すのがよろしい．データのセーブが終わると，「OK」と表示されてプログラムは終了する．

ところで，セーブが終わったら，その結果を必ず確認するという姿勢は，常に正しい．われわれとしても，ぜひ皆さんに実行してほしいと願うものだ．確認といったって，よーするに間違いなくロードできるかどうか一回試してごらんなさいってこと．

ここで作成したプログラムをロードするには，

```
LOAD"CAS:FONT" RETURN
```

とする．「CLOAD」ではなくて「LOAD」を使用しているのは，これがアスキーセーブされているからだ．このような，「プログラムによって作り出されたプログラム」は，すべてアスキーセーブになるのである．

さて，無事にロードが終了したら，

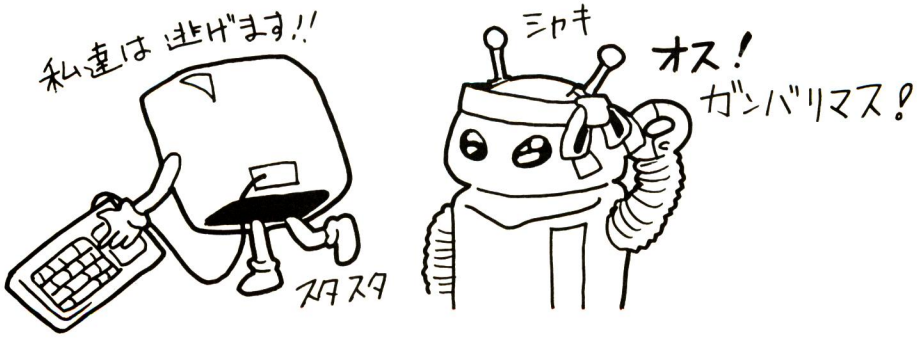
```
LIST RETURN
```

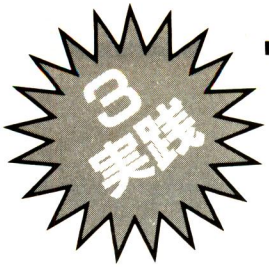
と入力して，内容を見てみよう．たとえば上と同じキャラクタ・フォントを登録した場合なら，77ページ〔リスト2〕のようなプログラムができていると思う．

いいですか，この4行のプログラムは，キミが手で打ち込むんじゃありませんよ．あくまでも，リスト1を実行した結果として自動的にテープ2にセーブされるべきものなんですよ．わかってもらえるかなあ．

● ゲーム本体(リスト3)の入力 ●

さて，お次はリスト3を入力してもらおう．ここが実際のゲーム進行をつかさどる部分である．この本全体をとおしてもっとも長いリストでもあるし，またわざと打ち込みミスが出やすくするような仕掛けも数ヵ所ほどこしてあるので，細心の注意をはらって打ち込んでほしい．





イジワル・リストに気をつけろ

ポケットバンク読者の皆さんからのアンケートをながめていると，バグというものは実に様々な場面に出現するものだ，と感心させられる．だが，大量のアンケートハガキの集計によって，「特にこの部分でバグが発生しやすい」という個所がしだいに浮かびあがってきた．

オカルト話が大好きなプログラマーのK氏は，このバグ多発地のことを「パソコン界のバーミューダトライアングル」と名付けたが誰にも相手にされず，ひとり酒場で泣いていたという．今の若い人は「魔の三角地帯」なんて知りませんよねエ．

……などという，センスのかけらもないウチワの話はさておき，ここではその「バグが発生しやすい時と場所」について，しばらく重点的に説明しよう．

●「コメント文」にコメントする ●

リスト3の最初の行（1000行）はREM文だけの行である．

REMは，先にもいったようにプログラムを見やすくするために書く文で，

```
1000 REM・・・・・・
```

のように「REM」で書き始める方式と，

```
1000 '・・・・・・・
```

のように，シングルクォート（ ' ）で書き始める方式の2とおりがある．

特に，このシングルクォート（ ' ）は小ちゃくて目立たないから，プログラムの見た目をすっきりさせるのに非常によろしい．そのため最近では「REM」よりも愛好者が増えている．

だが，なにしろ小さいから，印刷インクがたまたま運悪くそこでカスれてしまったりすると，まったく消えてしまうこともある．たとえば1010行（これは実は，

単に読者の頭が混乱するようにと思ってつけた，まったく意味のないコメントなんだけど)，この行のシングルクォートが消えて見えなくなってたら，どーしますか，はいアナタ？

考えられる対応のひとつは，すなおに，なにも考えずに，この行をシングルクォート抜きで打ち込み，気がつかないでRUNしてしまうこと．でも，

```
1000 FOR U = GOTO HELL! ----
```

なんてのは，「FOR文」でも「GOTO文」でもないデタラメ文だから，当然ここでエラー発生！　プログラムは止まってしまう．

プログラムにだんだん慣れてくると，「こういう命令は存在しないはずだ」なんていうことは，けっこう直感的にわかってしまう．「きっとこりゃー，シングルクォートが消えてるんだよ」なんて言って，涼しい顔でプログラム入力を続けられるように，早くなりたいもんですね．よーするに慣れの問題なんですが．

●「AUTO」殺すにゃ刃物はいらぬ，行番号の不ぞろいあればよい●

プログラムリストを打ち込む時に，AUTOコマンドを利用する人は多いと思う．あれは手間がはぶけていいもんらしいですね．だけど，プログラムがきちんと等間隔の行番号でつくられていればいいけれど，そうじゃない，ふとどきしごくなプログラムも，時々存在する．

そのようなプログラムを，そのようなプログラムとも知らずにAUTOコマンドを使って打ち込んでいると，最後にはGOTO文の行き先などがメチャクチャになってしまう．

プログラマーが途中でプログラムに改良を加えたとか，バグを取るなどして行番号がデコボコになってしまうことはよくあるものだ．でも，そのデコボコのまんまのプログラムを世間に発表するなんて，許せない．私は声を大にして言う．そのような恥ずかしいプログラムとは，つきあいたくないっ！

……ん？　あれれ，そんなこと言ってたら，どうも私の1200行近辺にイヤーな雰囲気を感じてきた……．おっとお，こりゃなんですか！　ここんところだけ行番号が1つ置きになってるじゃないの．えへン，いやぁね，おホン，こーゆーこともあるんですよ，たまには，あっはっは．

まあ，しょうがないですから，こうしてください．まず「AUTO　1000」を使って1190行まで打ち込む．そしたら今度は「AUTO　1200，1」を使って1208行まで打ち込む．あとは「AUTO　1210」を使って最後までいっきに打ち込む．面倒くさいけどそうしてね．

結論．AUTOコマンドを使用する時は，行番号の不ぞろいだけは気をつけましょーね．

● なにがなんでも〔RETURN〕キーを押してしまう画面の端 ●

例えば1020行を見てほしい．この行は，紙面や画面で見るかぎり，

```
1020 SCREEN 1,,0,1,0:KEY OFF:WIDTH 32:CO
```

というように右端が「……CO」で終わっている．

MSXにまだ慣れていない人は，よくここで〔RETURN〕キーを押してしまうけれど，それは大きな間違いですよ．次の行を，

```
LOR 15,4,4:CLS [RETURN]
```

ていう具合に打ち込めばエラーが出てきて一目瞭然．つまり，1020行を右端まで「……CO」と打ち込んだら，そのまま続けて「LOR……」と打って良いのである．

同じことなんだけど，もうすこし事態が複雑なのは1490行．これは1490行の端が「……THEN」ですっぱり切れていて，次に

```
1410 'caution
```

というコメントだけの行があるように，どうしたって見えてしまう．だけど，ちょっと待って．1490行の次に，なにゆえ1410行が現れるのか！

つまり，これは1490行の最後から，

```
···· THEN 1410
```

と続いている行番号だったのである.

1490行のように，最後が「THEN」で終わっていたり，「GOTO」，「GOSUB」で終わっている行は，だから要注意です.

● 数字に続いて現れる「小文字のl(エル)」,「大文字のO(オー)」 ●

小文字のl(エル)は数字の1と間違えやすく，大文字のO(オー)は数字の0と誤りやすい．そのことは十分わかっていても，やはりこんな場合にはついついミスを犯してしまうんじゃありませんかねぇ.

例えば1040行，この中にでてくる文字列は，どうしても

```
"v8164"
```

に見える．だけどこれは

```
"v8l64"
```

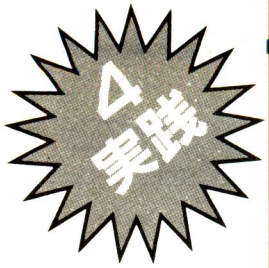
が本当．この違いを見分けるには，ある程度BASICの知識も必要になってくる．つまり，PLAY命令で使われる文字列の中で“V”というのは音量を表すのだから，後ろに8164（八千百六十四）なんて大きな数が続くはずはない！ したがって，これは“V8L64”のことだな，と判断するわけである.

それから1310行，この行はわざと字間をつめてあって見にくいのだが，決して

```
····32)>290 RI>735 THEN ····
```

ではない．BASICを少し知ってくると，上のようなプログラムは意味をなさないことがわかるものなんですよ．正しい1310行は，下のとおり.

```
····32)>29 OR I>735 THEN ····
```



いざ，実行！

いよいよ最終段階．こころはハヤるが，あせってはいけない．

● MERGEの実行 ●

リスト3が入力できたら，いよいよ最終段階．先に作っておいたキャラクタフォント・データのプログラム（リスト2）と，このリスト3をマージしよう．

まず，安全のために，今打ち込んだリスト3はセーブしておく．

```
CSAVE"GAME" RETURN
```

そうしたら，カセットレコーダーにキャラクタフォント・データ（リスト2）の入っているテープ（テープ2）をセットして，次の命令を実行する．

```
MERGE"CAS:FONT" RETURN
```

無事にマージが終了すれば，MSXの中には，みごと実行可能なゲームプログラムができあがっているという次第である．

おっと，ここでのんびりしてはいけなかったのだ．プログラムはまだMSXのメモリの中にいるだけだもの．スイッチを切ったらきれいに消えちゃいます．4本めのテープを用意して，しっかりとセーブしておこう．

```
CSAVE"DIGING" RETURN
```

あーあ，おつかれさまでした．

● ゲームの使用法 ●

最後にゲームの遊びかたなど，少々説明しておこう．

RUNすると，

```
    ﾄﾞﾉ ﾚﾍﾞﾙﾆ ｲﾀｼﾏｼｮｳ
```

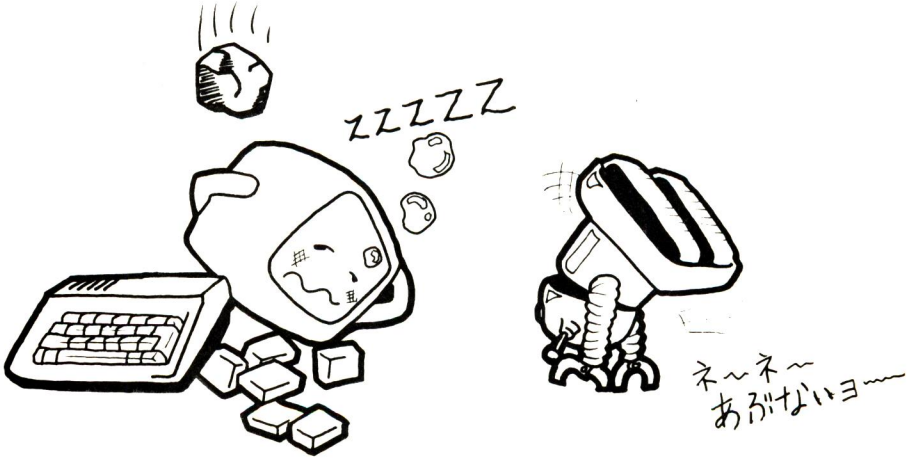
の質問がどーんとやってくる．1から9にレベルが増えるにしたがって，ゲームのむずかしさも増すからそのつもりで，レベルを決定しよう．

レベルを決めると，その面の岩の配置が画面に表示される．そして画面の最上行の左端に残り時間（初期値は200秒），同じく右端に得点（初期値は0点），同じく中央に赤いキャラクタが現れる．この赤いキャラクタは人間を表していて，カーソルキーで上下左右に移動できる．

この赤い人を動かして土を掘り，岩を落とせばいいわけだ．すでに書いたように，岩は空間があいていればどこまでも落ちていき，その落下距離が長いほど得点も多い．

こうして，200秒が経過するか，赤い人が四方を岩に囲まれて身動きとれなくなってしまうか，またはプレイヤーが〔ESC〕キーを押すとゲーム終了である．

9レベルのすべてにおいて1万点以上を得られれば，私はあなたをほめてあげましょう．では健闘をいのる．



[リスト3]

```
1000 '===== digging game ==============
1010 'FOR U = GOTO HELL! ...
1020 SCREEN 1,,0,1,0:KEY OFF:WIDTH 32:CO
LOR 15,4,4:CLS
1030 DEFFNW$(X,N)=RIGHT$(SPACE$(N)+STR$(
X),N)
1040 DEFFNPP$(T$)="v8164"+T$
1050 ST=0:PLAY FNPP$("t255")
1060 ON INTERVAL=60 GOSUB 1790
1070 DIM D(8)
1080 '
1090 FOR I=1 TO 9:READ D(I-1)
1100 DATA ,-32,,1,,32,,-1,:NEXT
1110 '
1120 C1=ASC(" ") : 'space
1130 C2=ASC("ｵ")   'earth
1140 C3=ASC("ｶ")   'stone
1150 C4=ASC("ｷ")   'wall
1160 C5=ASC("ﾌ") : 'man
1170 '
1180 VPOKE BASE(6)+C5/8,&H84
1190 V=BASE(7)+C2*8
1200 FOR I=0 TO 3
1201 FOR J=0 TO 7
1202 READ DO(I,J)
1203 NEXT J,I:READ M$
1204 I=0
1205 FOR N=I TO 3
1206 FOR M=I TO 7
1207 VPOKE V+N*8+M,DO(N,M)
1208 NEXT M,N
1210 '
1220 '--------------- screen init -----
1230 '
1240 CLS
1250 PRINT"  ﾄﾞﾉ ﾚﾍﾞﾙに いたしましょう (1-9)?";
1260 K$=INPUT$(1):IF K$<"1" OR K$>"9" TH
EN 1260
1270 LV=VAL(K$):I=RND(-LV):R=(20+LV)/80
1280 '
1290 CLS
1300 FOR I=32 TO 767
1310 IF ((I-1)MOD32)>29ORI>735THEN C=C4:
GOTO 1330
1320 IF RND(1)>R THEN C=C2 ELSE C=C3
1330 VPOKE BASE(5)+I,C
1340 NEXT I
```

```
1350 '
1360 '-------------- main routine -----
1370 '
1380 SC=0:LOCATE 29,0:PRINT SC
1390 P=BASE(5)+15:VPOKE P,C5
1400 TM=200:INTERVAL ON
1410 '
1420 IF INKEY$=CHR$(27) THEN 1670
1430 IF TM<=0 THEN 1670
1440 J=STICK(ST)
1450 IF J<>0 THEN W=8:O=J:GOTO 1480
1460 W=W-1:IF W>0 THEN 1410
1470 '
1480 Q=P+D(J):IF Q<BASE(5) THEN 1410
1490 V=VPEEK(Q):IF V<>C2 AND V<>C1 THEN
1410 ' caution
1500 '
1510 SWAP P,Q:VPOKE Q,C1:VPOKE P,C5
1520 IF VPEEK(Q-32)<>C3 THEN 1410
1530 '
1540 PLAY"v4o7gec":T=LV
1550 VPOKE Q-32,C1:VPOKE Q,C3:Q=Q+32
1560 IF VPEEK(Q)=C1 THEN T=T*1.5:GOTO 15
50
1570 '
1580 INTERVAL STOP
1590 SC=SC+INT(T)
1600 LOCATE 24,0:PRINT FNW$(SC,7)
1610 INTERVAL ON
1620 IF VPEEK(P-32)<C3 THEN 1410
1630 IF VPEEK(P+1)<C3 THEN 1410
1640 IF VPEEK(P+32)<C3 THEN 1410
1650 IF VPEEK(P-1)<C3 THEN 1410
1660 '
1670 INTERVAL OFF
1680 LOCATE 1,0:PRINT"GAME";" ";"OVER"
1690 IF PLAY(0) THEN 1690
1700 SOUND 7,&H37:PLAY M$
1710 IF PLAY(0) THEN 1710
1720 SOUND 7,&H3E
1730 IF INKEY$<>CHR$(13) THEN 1730
1740 '
1750 GOTO 1220
1760 '
1770 '----------- timer interrupt -----
1780 '
1790 LOCATE 0,0:PRINT FNW$(TM,4)
1800 TM=TM-1
1810 RETURN
1820 '
1830 '----- data -----
1840 '
```

COLUMN

プログラマたちの会話から

どこも間違ってないんだけどエラーが出る！？

```
Syntax error in 6800
```

「わっ，びっくりした．またエラーか．6800行に間違いがあるのか，便利だな，BASICは，こうして教えてくれるもんなあ．」

```
LIST 6800 RETURN
6800 IF FNA(I)=0 THEN PRINT"AHO"
Ok
```

「？？？　どこが違うのかなあ．わからないなあ．ねえ，Kさん，これどこが違うの？」

「これ？どこも違ってないなあ．」

「じゃ，このMSXこわれているの？」

「いや，そういうわけじゃないんだ．エラーのある（と示された）行に間違いがあるとは限らない．この場合だとね．このFNA(I)という関数に原因があるな．」

「え，関数？これ配列じゃないの？」

「そう，“FN”で始まる名前は，FN関数などと呼ばれ，特別な扱いになるんだ．プログラムのどこかに，必ずDEF FNというのがあるから見てごらん．ほれ，LIST．」

```
  ⋮
1020 DEF FNA(X)=SQR(X)*3+X*20
  ⋮
```

「ね，この1020行で，FNA（＜数値＞）というのは，＜数値＞のルートを3倍したものに，＜数値＞の20倍を足す関数であることを示している．これを関数の宣言とか定義などと言うんだ．」

「なるほど」

「で，だな．ここをよく見てごらん．20であるべきところが２０（二・オー）になっている．これじゃ，エラーになるなあ．」

「あっ，本当だ．よく見直したつもりなんだけどなあ．」

「こんな時は，関数を定義している1020行では，エラーにならず，それを参照した6800行でエラーになってしまうんだ．」

「ふーん．それはわかったけどさ，どうして1020行でエラーにしてくれないの？そのほうが便利なのに．」

「ギクッ，……パラパラ（本をめくる音）……ウ，ウォッホン．つまりだな，それは，その……BASICを作った人が，そのほうがいいと思ったんだよ．きっと．」

「あんまり，説明になっていないなあ……」

「このほかにも，エラーであると示された行に誤りのない場合は，けっこうある．たとえば，「Undefined line number（指定された行番号が見つからない）」は，その行が違っていることも，もちろんあるが，その行のGOTOやGOSUBで示された行を入れ忘れている，ということがある．」

「はあ，（このヤロー，話題をそらしたな）．」

「そのほかよくあるのは，PLAY AX$（1，2）とか，DRAW B$のように，文字変数のからんでいるところで，「Illegal function call」が出るというトラブルで，これは，質問の中でも，圧倒的に多い．これは，一言でいって，DATAの誤りだな．」

「あのですね．一言でいって，DATAの誤りだといいますが，データって，ほら，こんなにあるんですよ．どうやって，探すんですか．」

「この本のどっかに書いてあるよ．自分で探せ．」

「パラパラ……これですか（38P）．もっと簡単な方法はないんですか？」

「だから……ページも尽きたので，このへんで．」

「あっ，きったねー．」

おまけのプログラム
万能カセットFILES

FILES（ファイルズ）とは，プログラムの一覧表のこと．あるカセットにはいってるプログラムの一覧を表示してくれる，便利もののプログラムを紹介しよう．

●使い方

プログラムをRUNさせると現在読み込んでいるDATAの行番号を表示しながらマシン語をメモリー上においていく．もしDATA ERROR IN XXXXと表示して止まったらXXXX行をチェックしよう．DATAに誤りがなければ“MAC data ok”と表示してマシン語の実行に移る．

タイトルが表示されたら，カセットを巻き戻してから，再生ボタンを押す．カセットを全部読まないと一覧表は完成しないから，その間お茶でも飲んで待っていよう．表示は，最初がプログラム名，コロンの次がファイルタイプだ．ファイルタイプで，BASICと出ればCSAVEされたプログラム，ASCIIと出ればアスキーセーブされたプログラム，M−CODEなら，マシン語のプログラム．マシン語のファイルの場合は，開始番地と終了番地と実行開始番地も表示する．なお，プログラムを止めたいときは〔CTRL〕+〔STOP〕でストップする．

```
FILES FOR CAS:    BY K

GAME1  :BASIC
SUB    :M-CODE
 START=D000 END=E000 EXEC=D004
GAME2  :ASCII
```

まず，間違いのないようにこのプログラムを打ち込もう．マシン語プログラムだから，RUNさせる前には必ずセーブしておこうね．

```
100 '====== cas files ======
110 '            by kazuhi.k
120 '
130 CLEAR 100,&HD000:CLS
140 L=1000:DEFUSR=&HD000
150 FOR I=&HD000 TO &HD13F STEP 8
```

```
160    SU=0:FOR J=I TO I+7
170      READ A$:A=VAL("&H"+A$):POKE J,A
180    SU=SU+A:NEXT J
190    READ A$:A=VAL("&H"+A$)
200    IF A<>SU THEN PRINT "DATA ERROR IN
";L:END
210    LOCATE 10,0:PRINT "LINE";L
220 L=L+10:NEXT I
230 PRINT:PRINT"MAC DATA OK"
240 A=USR(0):END
250 '----- mac data -----
1000 DATA CD,7B,D0,3A,16,D0,5F,21,3B8
1010 DATA 17,D0,CD,6E,D0,CD,7B,D0,50A
1020 DATA CD,7B,D0,C3,8A,D0,15,46,490
1030 DATA 49,4C,45,53,20,46,4F,52,234
1040 DATA 20,43,41,53,3A,20,20,20,191
1050 DATA 42,59,20,4B,06,3A,42,41,1C9
1060 DATA 53,49,43,06,3A,41,53,43,1F6
1070 DATA 49,49,07,3A,4D,2D,43,4F,1DF
1080 DATA 44,45,08,20,20,53,54,41,1B9
1090 DATA 52,54,3D,05,20,45,4E,44,1DF
1100 DATA 3D,06,20,45,58,45,43,3D,1C5
1110 DATA CD,E1,00,1E,17,21,3B,D1,310
1120 DATA D5,E5,CD,E4,00,E1,D1,D8,5F5
1130 DATA 77,23,1D,20,F3,C9,7E,D5,3E6
1140 DATA E5,CD,A2,00,E1,D1,23,1D,446
1150 DATA 20,F4,C9,3E,0D,CD,A2,00,397
1160 DATA 3E,0A,CD,A2,00,C9,CD,E7,434
1170 DATA 00,C9,CD,58,D0,DA,86,D0,4EE
1180 DATA 3A,3B,D1,FE,D3,CC,A4,D0,557
1190 DATA FE,EA,CC,BA,D0,FE,D0,CC,6D8
1200 DATA D0,D0,18,E6,1E,06,21,45,328
1210 DATA D1,CD,6E,D0,3A,2C,D0,5F,471
1220 DATA 21,2D,D0,CD,6E,D0,CD,7B,471
1230 DATA D0,C9,1E,06,21,45,D1,CD,3C1
1240 DATA 6E,D0,3A,33,D0,5F,21,34,32F
1250 DATA D0,CD,6E,D0,CD,7B,D0,C9,5BC
1260 DATA 1E,06,21,45,D1,CD,6E,D0,366
1270 DATA 3A,3A,D0,5F,21,3B,D0,CD,39C
1280 DATA 6E,D0,CD,7B,D0,3A,42,D0,4A2
1290 DATA 5F,21,43,D0,CD,6E,D0,ED,48B
1300 DATA 4B,4C,D1,CD,1C,D1,3A,4B,3A7
1310 DATA D0,5F,21,4C,D0,CD,6E,D0,477
1320 DATA ED,4B,4E,D1,CD,1C,D1,3A,44B
1330 DATA 51,D0,5F,21,52,D0,CD,6E,3FE
1340 DATA D0,ED,4B,50,D1,CD,1C,D1,4E3
1350 DATA CD,7B,D0,C9,1E,04,16,04,31D
1360 DATA 3E,00,CB,11,CB,10,CB,17,2D7
1370 DATA 15,20,F7,C6,30,FE,3A,FA,454
1380 DATA 34,D1,C6,07,CD,A2,00,1D,35E
1390 DATA 20,E4,C9,00,00,00,00,00,1CD
```

トラブル リファレンス

エラー撃退ミニ事典　　MSXポケットバンク16

1985年1月25日　初版発行　定価480円
編著者　ポケットバンク編集部
発行者　塚本慶一郎
発行所　株式会社アスキー
〒107　東京都港区南青山5－11－5住友南青山ビル5F
振　替　東京4-161144
電　話　03－486－7111（代表）

編集担当　八木淳一
表紙担当　郷　啓子
印刷　株式会社加藤文明社印刷所

ISBN4-87148-772-5 C0055 ¥480E

MSX POCKET BANK

音楽好き、
ゲーム好き、
アート好き。
それぞれうれしい
MSXポケットバンク。

各巻480円

1 アニメC.G.に挑戦！
2 マイコン・ジュークボックス
3 BASICゲーム教室
4 マイコン・サウンドパック
5 ゲームキャラクタ操縦法
6 トランプゲーム集
7 面白パズルブック
8 プログラムD.J.
9 グラフィック秘伝
10 マイコン野球中継'84

11 とにかく速いマシン語ゲーム

とりえは速さだけではない。
面白さバツグンのマシン語
ゲームがキミを直撃！

ポケット
バンク
編集部

12 アクションゲーム38

あっ、という間に打ちこめるオ
モシロゲームが38!!
ポケットバンクの出血大サー
ビス。

ぐるーぷ・
アレフ

13 知能ゲーム38

キミの頭は噴火寸前。
楽しく悩める知性派ゲームが、
38本も直撃だ。

ぐるーぷ・
アレフ

14 必殺ビデオ活用法

ビデオとMSXつなげたら、
10倍広がるビジュアル・ワー
ルド。今、君だけの映像を、
手に入れろ!!

ポケット
バンク
編集部

15 占っちゃうから！

世の果てからやって来た、不
可思議占い！
おばっちゃん、おじょうちゃ
んには、過激すぎるぜ。

ポケット
バンク
編集部

16 エラー撃退ミニ事典

迷える小羊に救いの道を・・・
プログラマのバグとり術を満
載。
君の悩みも、これで解決。

ポケット
バンク
編集部

■全国有名書店、マイコンショップでお求めください 〒107東京都港区南青山5-11-5住友南青山ビルPHONE 03(486)7111代 株式会社アスキー

MSX POCKET BANK

MSXポケットバンク⑯

エラー撃退ミニ事典

CONTENTS

MSXポケットバンク・シリーズ
好評既刊　各巻480円

①アニメC.G.に挑戦！
②マイコン・ジュークボックス
③BASICゲーム教室
④マイコン・サウンドパック
⑤ゲームキャラクタ操縦法
⑥トランプゲーム集
⑦面白パズルブック
⑧プログラムD.J.
⑨グラフィックス秘伝
⑩マイコン野球中継'84
⑪とにかく速いマシン語ゲーム集
⑫アクションゲーム38
⑬知能ゲーム38
⑭必殺ビデオ活用法
⑮占っちゃうから！
⑯エラー撃退ミニ事典

ISBN4-87148-772-5　C0055　¥480E

定価480円