

```

710 PRINT "イイデスカ [Y/N] ?"
720 GETG$: IF G$="Y" THEN 30
730 GOTO 720
740 REM ** フリントー ***
750 PRINT:PRINT "フリントー ニ シュウリヨウ チュウ"
760 FOR J=1 TO I
770 PRINT/P USING "###";J; "[NAME] ";AA$[0,J]:PRINT/P "[TEL.]";AA$[1,J]
780 PRINT/P " ";AA$[2,J]
790 PRINT/P
800 NEXT J
810 PRINT "オワリ":MUSIC "ABC"
820 GOTO 30

```

"イイデスカ[Y/N]?"を画面に出す。
キーボードで"Y"が押されたら行番号30へとぶ。
行番号720へとぶ。
注釈文。
1行改行し、「プリントに出力中」と画面に出す。
Jが1からIの間、行番号800までを繰り返す。

プリントにJの値と"[NAME]"と名前を書く。
プリントに"[TEL.]"と電話番号を書く。
プリントに住所を書く。
プリントの改行をする。
行番号760の繰り返し命令の終りを示す。
"オワリ"と画面に出す。音を出す。
行番号30へとぶ。

[解説]

WOPEN/T、ROPE/T、CLOSE/T、PRINT/T、INPUT/Tといった命令を用いてカセットテープにデータ・ファイルを作成します。このプログラムは100人以内の住所録を作成しテープに記憶したり読みだしたりできます。最初に何をするかを聞いてきますので1~6の数字で答えます。そして、それぞれの処理をするようになっています。また、行番号690のUSR(\$00D8)はモニターの内部サブルーチンを呼ぶ命令で、GET文による連続入力を可能にしています。

1 ADDRESS ノ サクセイ

入力終わりの時は、NAMEの時Eを入力してください。また、まちがえて1 ADDRESS ノ サクセイになってしまった時は、NAMEの時Mを入力してください。名前、電話番号、住所の入力まちがいは全部入力した後で2ヘンコウを選んでそれから変更してください。

2 ヘンコウ

まちがえて2ヘンコウになった時や住所録の変更が終った時は、NAMEの時Eを入力してください。住所録の変更は変更するNo.を入力してから行います。

3 カキコミ

カセットテープにデータファイルを作成します。

4 ヨミトリ

カセットテープからデータファイルを読み込みます。

5 ミル

住所録を見る。スペース・キーを押すと止まります。

6 プリントーへ カク

プリントーに住所録を書きだします。

```

*** ADDRESS PROGRAM ***
1 サクセイ   2 ヘンコウ
3 カキコミ   4 ヨミトリ
5 ミル      6 フリントーへ カク
トドレニ シマスカ (1-6)

```

```

ADDRESSノサクセイ<オワリ=E, マチカニイ=M>
NO. 1 NAME BANDAI
      TEL. 03-842-5151
      ADDRESS KOMAGATA
NO. 2 NAME BANDAI EL.
      TEL. 03-842-5155
      ADDRESS KOMAGATA
NO. 3 NAME BANDAI KAIJIN
      TEL. ■

```

データファイルは?

WOPEN/T

使用例

WOPEN/T "データ1"

データ1というデータファイルを作るために、カセットデータファイルをオープンする。

一般型式 WOPEN/T "データファイルの名前"

説明 ●プログラムで使うデータの集まり（データファイル）をカセットテープに記録したり、逆に読み込んだりするするために、カセットテープを準備（カセットデータファイルをオープン）する。

WOPEN

- データファイルをカセットテープに作成するために、カセットテープを準備（カセットデータファイルをオープン）する。
- データファイルの名前は16文字までで、省略してもよい。ただし、1本のカセットに複数のデータファイルをもつ場合は、データファイルに名前がついていないと不便であり、間違いをおこしやすい。
- WOPEN/T命令を実行すると、画面に
WRITING [ファイルの名前]
となる。
- データファイルを作成し終ったならば、CLOSE/T命令を実行しなければならない。これを実行しなかった場合には、不完全なデータファイルがカセットテープにできてしまい、このデータファイルを読み込んだとき、あるデータ以降が正しく呼び出せないということが生じてしまう。

データファイルを作成するときに、各命令名は、最初頭に記入されている。たとえば、OPEN/Tは「データ」、CLOSE/Tは「データ」、DATA/Tは「データ」という形式のデータファイルとしてカセットテープに記録される。

データファイルは? ROPEN/T,CLOSE/T

使用例

① ROPEN/T "データ1"

データ1というデータファイルを読み込むために、カセットデータファイルをオープンする。

② CLOSE/T

データファイルを閉じる。

一般型式 **ROPEN/T** "データファイルの名前"
CLOSE/T

説明 ROPEN/T

- カセットテープに記録してあるデータファイルをRX-78に読み込むために、指定した名前のデータファイルを準備(オープン)する。
- データファイルの名前を指定しなかったならば、1番はじめに見つけたデータファイルをオープンする。
- データファイルを読み終わったならば、CLOSE/T命令を実行してファイルを閉じる。

CLOSE/T

- WOPEN命令を実行している場合、オープンしているデータファイルをクローズして、カセットテープ上に1つのデータファイルを完成する。
- ROPEEN命令を実行している場合、オープンしているデータファイルをクローズして、新たに別のファイルが使用できる(オープンできる)ようにする。

データをカセットに書きこめ! -PRINT/T-

使用例

PRINT/T A\$, A 変数A\$, Aのデータを、カセットテープに書きこむ。

STEP 8 サンプルプログラム

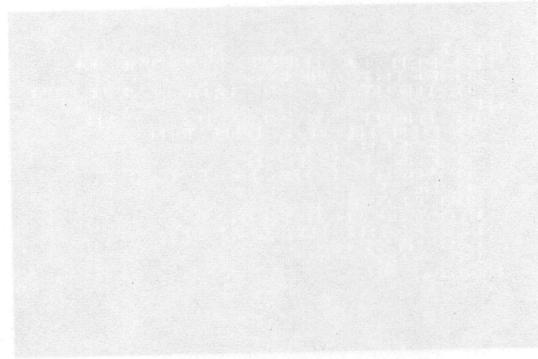
一般型式 **PRINT/T** 数値データまたはストリングデータ
, 数値データまたはストリングデータ

説明 ●WOPEN命令によってオープンしているカセットデータファイルに、データを順次書きこんでいく。

●数値データ、ストリングデータは数値変数、文字変数あるいは配列などで与えてもよい。また、1つのPRINT/T命令に2個のデータを置く場合は、コンマ(,)で区切ればよい。

●PRINT/T命令実行中にプログラムを中断してしまったり(エラーやSHIFT+STOPなどによる中断)、CLOSE/T命令を実行せずにいた場合には、不完全なデータファイルになってしまふ。

PRINT/T A\$, A



```
LIST REM ** PRINT/Tサンプル ***
20 PRINT "TEST"
30 INPUT "ナンバーを入力"; N
40 DIM H$(N); DIM T(N)
50 FOR I=1 TO N
60 INPUT "タマゴ?"; H$(I)
70 INPUT "テンスウ?"; T(I)
80 NEXT I
90 WOPEN/T "テスト.ケッカ.1"
100 FOR J=1 TO N
110 PRINT/T H$(J), T(J)
120 NEXT J
130 CLOSE/T
140 END
Ready
```

行番号30~70で配列N\$と配列Tに、各自の名前と点数結果をまず入れている。いったん配列に入れられたデータは、行番号80~120で「TEST.ケッカ.1」という名前のデータファイルとしてカセットテープに記録される。

カセットのデータを読む INPUT/T

使用例

INPUT/T A\$, A

カセットテープのデータを、変数 A\$, A に読み込む。

一般型式 INPUT/T 变数, 变数

説明

● ROPEN命令によってオープンしているカセットデータファイルから、変数にデータを順次読み込んでいく。

● データの読み込みを 2 個同時に行う場合は、コンマ (,) で区切ればよい。ただし、変数の型とデータの型 (数値かストリングか) が合わなければならない。

● データファイルの名前を指定しなかったならば、1番目に見つけたデータファイルをオープンする。

CLOSE/T

● WOPEN命令を実行している場合、オープンしているデータファイルをクローズして、カセットデータは新しい一つのデータファイルを完成する。

● ROPEN命令を実行している場合、オープンしているデータファイルをクローズして、新たに別データが使用できる（オープンできる）ようになる。

```

LIST REM ** INPUT/T サンプル **
20 PRINT CHR$(6)
30 INPUT "データファイル名 ナメ ?"; N$
40 INPUT "データ名 カゼット名 ?"; T$N
50 DIM N$(N); DIM T$(N)
60 ROPEN/T A$N
70 FOR I=1 TO N
80 INPUT/T N$(I), T$(I)
90 NEXT I
100 CLOSE/T
110 FOR J=1 TO N
120 PRINT N$(J), T$(J)
130 NEXT J
140 END
Ready

```

行番号 60~90 で、指定したデータファイルが配列 N\$ と配列 T にデータが読み込まれ、行番号 100 でファイルを閉じている。配列に読み込まれたデータは、行番号 110~130 で画面に書きだされる。

STEP 6 マシン語へチャレンジ

RX-78のBS-BASICを全てマスターした君へ。

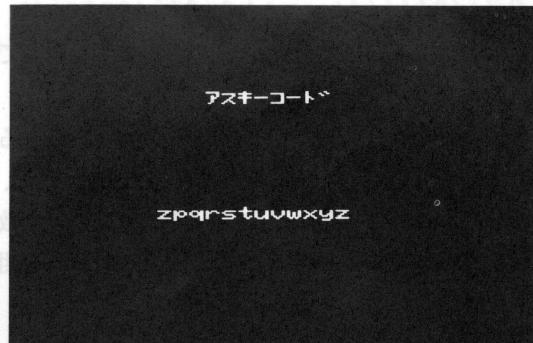
次のステップ、マシン語が待っている。これからは、BS-BASICの助けは期待できない。RX-78が突然応答しなくなったり、プログラムが消えてしまったりしてしまう。自分自身でRX-78のCPU Z-80のことによく学んでから挑戦しよう。

STEP 6 サンプルプログラム

```
10 REM ** アスキーコード ** ..... 注釈文  
20 PRINT CHR$(6) ..... 画面に出てる内容をすべて消す。  
30 CURSOR11,4:PRINT "アスキーコード" ..... 座標(11,4)に"アスキーコード"を画面に出す。  
40 FOR J=16 TO 120 STEP 12 ..... Jの値が16から120の間、行番号150までを繰り返す。  
50 FOR L=J TO J+10 ..... Lの値がJからJ+10の間、行番号140までを繰り返す。  
60 CURSOR 8,12 ..... カーソルを(8,12)に移動する。  
70 FOR I=L TO J+10 ..... Iの値がLからJ+10の間、行番号90までを繰り返す。  
80 PRINT CHR$(I); ..... アスキーコードIを画面に出す。  
90 NEXT I ..... 行番号70の繰り返し命令の終りを示す。  
100 FOR I=J TO L ..... Iの値がJからLの間、行番号120までを繰り返す。  
110 PRINT CHR$(I); ..... アスキーコードIを画面に出す。  
120 NEXT I ..... 行番号100の繰り返し命令の終りを示す。  
130 FOR K=1 TO 10:NEXT ..... Kの値を1から10まで変化させる。  
140 NEXT L ..... 行番号50の繰り返し命令の終りを示す。  
150 NEXT J ..... 行番号40の繰り返し命令の終りを示す。  
160 END ..... プログラムの終りを示す。
```

【解説】

アスキーコードのコード16からコード122までの記号や英数字を12個づつネオンサインのように表示しています。アスキーコードは、ある程度順番に並んでるので、FOR～NEXTをうまく使うと、とてもおもしろいプログラムができます。



* サンプルプログラム中に出てくる〔 〕はキーボード上で（ ）と同じです。

機械語(マシン語)とは何だろう?

BASICと機械語、どうちがう?

今までBASICについて、いろいろ勉強してきたが、わかったかな? パソコンには、その他にもいろいろな言語があるが、RX-78では、機械語も使える。

BASICというのは、人間が使っている数式や、英語などと、非常に近い形で表現する言語なので、プログラムが組みやすい。ところがRX-78は、BASICのプログラムをそのまま理解してくれるのではない。RX-78は、あらゆる命令、データを2進法すなわち、0と1の組み合わせのみによって、理解し、実行するのである。たとえば、BASICで、PRINT ABCと入力しても、すべて、0と1の機械語命令に翻訳されて実行されるのだ。

しかし、人間が直接この2進法でプログラムを作ることは、非常に困難であり、不可能に近い作業である。君だって、0と1だけが何百もならんでいるのを見ると、頭が痛くなってくるだろう。

そこで、人間にわかりやすく、組みやすいプログラム言語が考え出されたのである。RX-78では、BASICが使えるわけだ。

機械語でRX-78と対話する

それでは、RX-78が直接理解できるような言語で、プログラムを書くことはできないのだろうか。

そういう条件を満たしてくれるのが機械語である。これは、0と1の2進数でなく、0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,Fを用いて、16進数で表わす。(10~15までを1文字で表わす数字がないためA~Fで代用しているんだ。)16進数でプログラムを組むことも、かなり大変な作業だが、2進数で組む程のことはない。

INPUT ABC

STEP 6 セットアップ

```
10 REM
20 PRINT CHR$(-15)
30 CURSOR 1,15
40 FOR I=18 TO 19 STEP 15
50 FOR J=17 TO 18 STEP 15
60 PRINT CHR$(I)
70 PRINT CHR$(J)
80 NEXT I
90 FOR I=17 TO 18 STEP 15
100 PRINT CHR$(I)
110 NEXT I
120 FOR K=1 TO 10:NEXT K
130 END
```

(略)

BASICモードとモニタモード

では、RX-78に、機械語のプログラムを組む時はどうすればよいのだろうか。まず最初に、RX-78に、今から機械語で入力するぞと宣言してやらなくてはならない。その時使うのがMONというコマンドである。これを入力すると、BASICモードから、モニタモードに変わる。モニタモードになると、はじめて機械語のプログラムを入力できるのだ。

モニタモードの機能は、RX-78のメモリの内容を表示したり、書きかえたり、機械語の命令を外部から読み込んだり、機械語のプログラムを実行したりすることができる。

すなわち、モニタとは、機械語を作るときに最低限必要な道具なのだ。

文字1文字は、RX-78の内部では1バイト

2進数や、16進数の話が出てくると、よくおめにかかることばにバイトというのがある。バイトとは何であろうか。

2進数は、0と1の組み合わせだという話をしたが、この0とか1、ひとつづつをビットというのだ。

この0と1を8つ組み合わせて1バイトという。

1バイトを使うと、 $2^8 = 256$ 個までの文字を表現できるから、RX-78で使う文字としては十分なのである。

ところが、2進数は、幅をとりすぎるし、わかりにくないので、2進数と1:1対応している16進数を用いて表現する。よって、16進数は、1バイト=2桁の16進数ということになる。

2進数と16進数の対応表

10進数	2進数	16進数	10進数	2進数	16進数
0	00000000	00	13	00001101	0D
1	00000001	01	14	00001110	0E
2	00000010	02	15	00001111	0F
3	00000011	03	16	00010000	10
4	00000100	04	17	00010001	11
5	00000101	05	18	00010010	12
6	00000110	06			:
7	00000111	07			:
8	00001000	08			:
9	00001001	09	252	11111100	FC
10	00001010	0A	253	11111101	FD
11	00001011	0B	254	11111110	FE
12	00001100	0C	255	11111111	FF

文字を16進数で表わすキャラクタコード

BASICでプログラムを組むとき、Aという文字も、Aのままで入力されるわけではない。文字の1つ1つに、16進数の数字が、1:1対応で決まっている。その16進数を使って文字を表わしているのだ。例えば、学校で、出席番号ということがある。平均点を求めるプログラムなどで、名前を入力するかわりに、出席番号で入力したりする。これでもかまわないわけである。それは名前と出席番号が1:1に対応しているので、11番の人というのがわかれば、ああ加藤君だとわかるようになっているからである。

それと同じように文字にも出席番号が付いていて、RX-78は出席番号で文字を覚えている。

この出席番号をキャラクタコードという。（アスキーコード表を参照158ページ）。

BASICのメリット、機械語のメリット

今まで述べてきたように、BASICというのは、とても組みやすく、見やすいプログラム言語である。ところが、機械が直接よみ込むわけではないために、機械の性能を十分に生かせないというデメリットもある。それに対して、機械語は、直接読むことができるので、スピードも早く、機械の性能も十分生かせるが、プログラムが組みにくく、見にくいというデメリットがある。

ただ、機械語というのは、書きにくさのために、敬遠しがちであるが、これで1度プログラムを組んでみると、RX-78の仕組がとてもよくわかったような気になる。機械語でRX-78とツーカーの仲になる。

それぞれの長所短所を生かして、プログラムを書くときに言語を選ぶようにしよう。

BASICと機械語で、RX-78をフルに活用していこう！

BASICから機械語をコントロールする

今まで、BASICと機械語の切り替えを学び、それぞれの使い方を勉強してきたが、双方の長所を生かして、BASICプログラム中に、機械語をとり入れたりすることもできる。

BASICモードで、特定のアドレスのデータをのぞいて見ることができるのだ。

ちょっとのぞき見

PEEK, POKE

使用例

① PEEK (49450)

10進番地49450にはいっているデータを10進数になおす。

② POKE 49450, 175

10進数番地49450にデータ175(10進表現)をセットする。

③ POKE AD, DA

変数ADで指定する番地に変数DAで示される値をセットする。

一般型式

PEEK (指定アドレス)

POKE 指定アドレス, データ

説明

●RX-78のメモリと直接データのやりとりを行う命令である。

PEEK

●メモリアドレスを10進数又は、16進数で指定すると、そのアドレスの内容を与えてくれます。

POKE

●10進または16進データを指定するアドレスにセットする。

```
LIST REM ** PEEK, POKE **
1000 REM
1001 PEEK<49450>
1002 PRINT "A=";A
1003 POKE 49450,A+1
1004 B=PEEK<49450>
1005 PRINT "B=";B
1006 END
Ready
RUN
B=200
B=201
Ready
```

行番号20でアドレス49450番地のデータを変数Aに10進数に直し入れている。

次にAに1を加え、もとの番地に行番号40に戻し、ふたたび行番号50で、取り出してみた。

変数A, Bともに同じアドレスから取り出したことになるが、行番号40のPOKE命令で、データの値が変わったのがわかる。

コントロールを移す

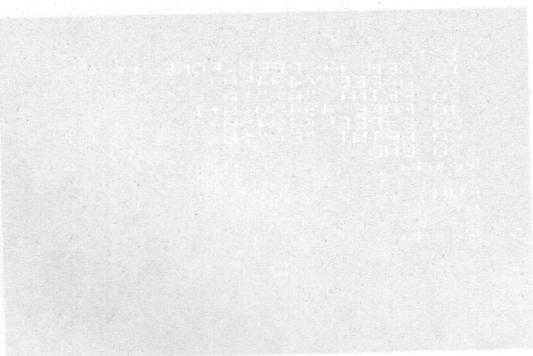
USR

使用例

- ① **USR (49152)** 10進番地49152にプログラムのコントロールを移す。
- ② **USR (AD)** 変数ADで指定させる10進番地にコントロールを移す。
- ③ **USR (\$C000)** 16進番地C000へコントロールを移動する。

一般型式 **USR (指定アドレス)**

説明 ●指定したアドレスにプログラムのコントロールを移す。
このコントロール移動は、機械語のCALLコマンドと同じ機能を持っている。したがって、機械語プログラムに、RETコマンドがあると、BASICプログラムへコントロールが戻る。



※モニターサブルーチン、Z80の命令に関しては取扱説明書を参照してください。

メモリの制限

LIMIT

使用例

① LIMIT 49151

BASICプログラムで使用するエリアを49150番地に制限する。

② LIMIT A

BASICプログラムで使用するエリアを変数A-1の番地に制限する。

③ LIMIT \$C000

BASICプログラムで使用するエリアを、16進番地BFFFに制限する。

④ LIMIT MAX

BASICプログラムで使用するエリアをメモリの最大にもどす。
(\$FFFF)

一般型式 **LIMIT** 制限番地アドレス

説明 ●BASICプログラムで使用できるのは、\$B000番地～\$DFFF番地となっている。(メモリマップ 160ページ参照) この使用エリアを制限するコントロール文がLIMITである。

●LIMIT \$C000と指定すると、\$B000～\$BFFFを使用可能なエリアとするということになる。

モニターの門

MON

使用例

① MON

RETURN

BASICモードから、モニターモードにかえる。

一般型式

MON

RETURN

説明 ●システムのコントロールを、BASICモードからモニターモードに移す。

●モニターモードに移ると、行の先頭に*が出る。

●モニターモードでは行番号は使用せず、すべてコマンドレベルで実行される。

●モニターモードからBASICモードに戻るには、モニタコマンドJ (145ページ) によって行なう (電源を切つたり、RESETしても自動的にBASICモードに戻る)。

Mコマンドでデータの変更

M

使用例

*ME011

E011番地からのデータを変更する。

一般型式

M 指定アドレス **RETURN**

説明

●アドレスを指定することにより、そのアドレスのデータを変更することができる。

●Mコマンドを実行すると、画面に指定アドレスのデータが表示され、修正する場合には、そのデータの右(カーソルがある位置)から正しいデータを書き、**RETURN**で入力する。

修正しない場合にはそのまま**RETURN**を押す。すると次のアドレスのデータが画面に出される。

●Mコマンドからぬけ出すには**STOP**を押す。

●\$CF, \$D7, \$DF, \$F7, \$FFをデータとして入れ、プログラムがこれを実行すると、ディスプレイにAF, BC, DE, HL レジスタ及びこのデータのあるアドレス、スタックポインタを表示し、コントロールをモニタコマンド待ちに戻す。これによって、デバッグするときのブレークポイントを設定することができる。

Dコマンドでデータを表示

使用例

*DE012 E022

E012番地からE022番地までのデータを画面にだす。

一般型式

D 開始アドレス 終了アドレス

スペース

説明

●開始アドレス～終了アドレスまでのデータを画面に出す。

●表示は1行8データずつ行なわれ、表示を途中で一時止めるにはスペースキーを押す。

●データを変更することはできない。

④開始アドレスと終了アドレスとの間には1つの空白(スペース)を入れて指定する。

Jコマンドでひとつ飛び

使用例

*JE012

E012番地へコントロールを移す。

一般型式

J 飛び先アドレス RETURN

説明

- 指定したアドレスにコントロールを移す。
- BASICモードにもどりたいときに、*J2005を指定する。(ベーシックプログラムは保存されています)。
- それ以外を指定すると初期状態(リセットを押した時)になります。

Lコマンドでファイルをロード

使用例

*L RETURN
FILE NAME : A

カセットテープにセーブされている機械語プログラムファイルネームAをRX-78にロードする。

一般型式

L RETURN

説明

- 指定するファイル名をもつ機械語プログラムファイルのローディングを行なう。
- ローディングを行なう場所は、Sコマンドで指定したアドレスからロードされる。
- ロードが正常に終了した場合、JUMP ADDRESSにジャンプする。
- チェックサムエラーが検出された場合、エラー表示を行ない、モニタコマンド待ちの状態になる。

Vコマンドでファイルの確認

使用例

*V RETURN
FILE NAME : A

カセットテープにセーブしてある
データファイル名Aが、メモリブロ
ック上の機械語プログラム、デー
タと同じものであるか確認する。

一般型式

V RETURN

説明

- カセットテープにセーブされた機械語プログラム、デ
ータが、そのもののメモリ・ブロック上の機械語プロ
グラム、データと同じであるかどうか調べる。
- ファイル名を指定しないと、最初に見つかった機械語
プログラムファイルについてベリファイを実行する。
- ファイルデータとメモリ・ブロック上のデータが同一
であれば"OK"が表示され、異なれば"ERROR"が
表示される。

Sコマンドでメモリをセーブ

使用例

*S RETURN
FILE NAME : A

指定するメモリブロックの内容を、
カセットテープにファイルネーム
Aとしてセーブする。

一般型式

S RETURN

説明

- 指定するメモリブロックの内容を、ファイルネームを付けてカセットテープ上に記録する。
- ファイル名は、16文字以内で指定する。16文字を越えた分は、無効となる。
- 次に、START ADDRESS、END ADDRESSを指定する。
- JUMP ADDRESSを指定すると、ローディングをした時、ローディングの後、JUMP ADDRESSへジャンプを実行する。
- JUMP ADDRESSを指定しない場合は、モニターのホット・スタート・アドレスが指定される。