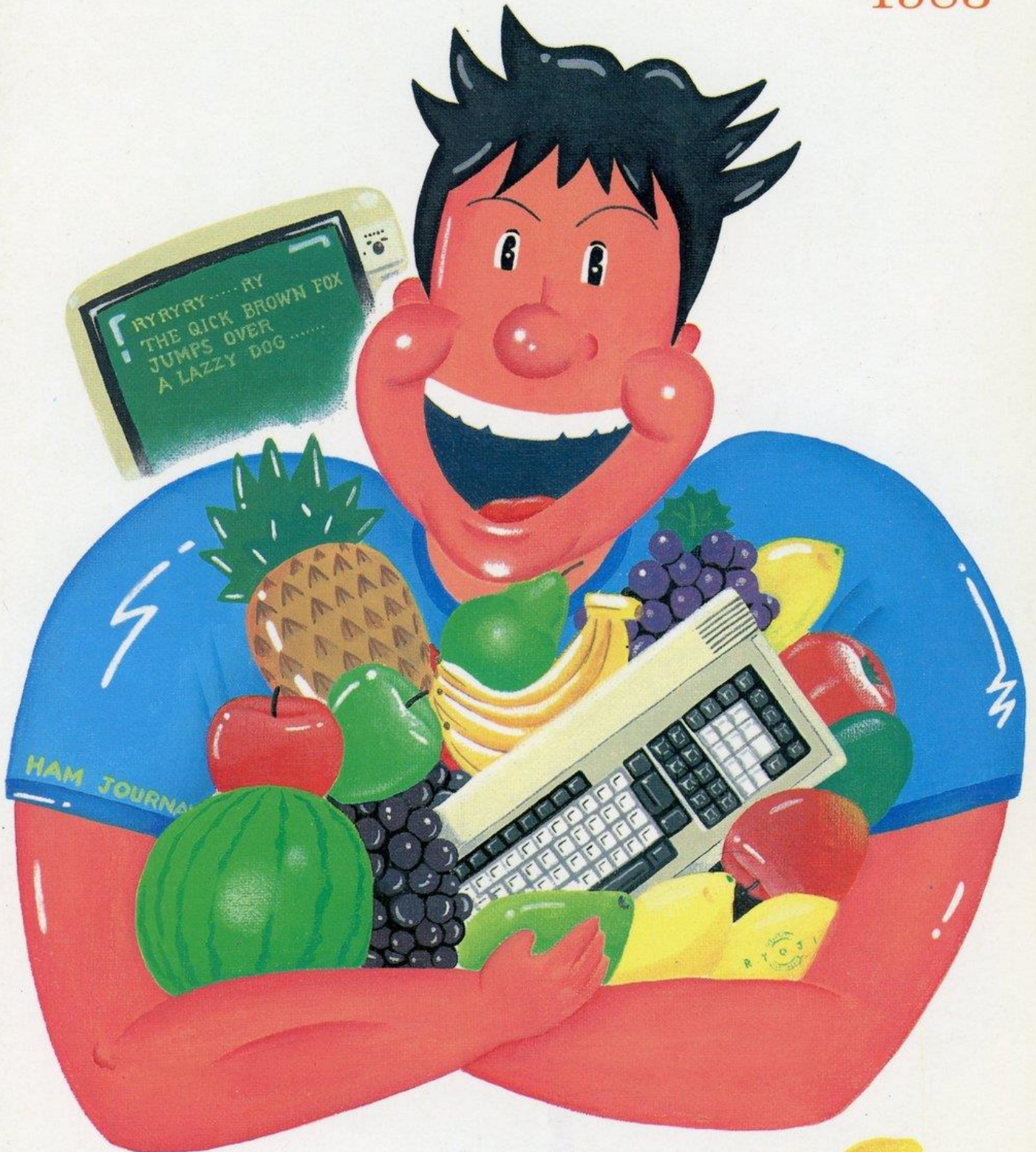


# 別冊 *HAM Journal*

ハム用パソコン・プログラム 1

1983



CQ出版社

# 別冊 *HAM Journal* CONTENTS

## ハム用パソコン・プログラム

- MZ-80K(E2)/-1200対応
- 2 多機能RTTYプログラム  
.....JA3ATR 小山 保昭
  
- PC-8001対応
- 27 機械語によるRTTYプログラム  
.....JK1PEC 大澤 一秋
  
- PC-6001対応
- 37 RTTY基本システム  
.....J11TQH 菱倉 仁
  
- MZ系SP-5030 BASIC対応
- 42 モールス符号送受信練習プログラム  
.....J13BVG マギー安岡
  
- PHC-25対応
- 56 CWキーヤー・プログラム  
.....JA3ESC 高杉 恒平
  
- MZ-80K(C)/-1200対応
- 62 BASIC CWトレーナー・プログラム  
.....JH3UWZ 龍野 一男
  
- PC-6001対応
- 74 コールサイン検索プログラム  
.....JF1XZD 大平 芳孝

# MZ-80K(C)/-1200対応

## BASIC CWトレーナー・プログラム

J H 3 U W Z

龍野 一 男

(JARL六甲CW愛好会)

### はじめに

今回、私たち六甲CW愛好会では、“HOW TO CW QSO”に次ぐ、CWビギナー向け第2弾として、パソコンを使ったCW練習機、すなわちモールス符号の受信練習ということでプログラムしてみました。ビギナー向けとはいいましたが、OMの受信練習として、和文も使用することができます。

ところで、最近のパソコン・ブームのおかげで、アマチュア無線家の中にも、パソコンを応用する例が増えてきました。CWの分野でも、もはやノー・オペレーターで局の運用、管理といったことが可能になりつつあります。私たちのように手動電信を愛する者にとっては、寂しい気もしますが、省力化を図るのは人間の常でもあり、しかたありません。そういう意味では、パソコンも諸刃の剣といえるでしょう。しかし、今回のような使い方をしたときには、本当に便利な先生になってくれます。長時間の使用にも不平一つもらしませんので、まったくFBです。パソコンは持っているが、CWはまだ知らない、これからやってみよう、というかたには、ぜひともこのプログラムを走らせ、そして手動電信にQRVしていただきたいと思います。

### 使用機種および言語

使用機種については、種々検討したのですが、“PC”を使ったプログラムは、先頃のCQ ham radio誌で発表されたことすし、クラブ内でも多くの人が使っており、たくさんの方が持っていると思われる、MZ-80K/C、1200を対象にしました。MZ-700についても使用可能ですが、これについては後述いたします。

さて、機種が決定したところで使用言語ですが、予定では、アセンブラを計画したのですが、とにかく一度プ

ログラムを入力して、走らせてほしいということからBASICにしました。今回使ったBASICはSP-5030ですが、CURSOR文を使っている関係で、旧バージョンのSP-5020では、若干のプログラムの訂正が必要となります。前述したMZ-700については、SP-5030 BASICをお持ちのかたは、まったく問題なく走らせることができます。

電源ON後、モニター1Z-009Aの起動後、SP-5030をロードして、プログラムを入力してください。なお、S-BASIC、Hu-BASICも同様ですが、カラー命令等がありますので、カラー化されるのもいいかと思います。

### 設計目標

このプログラムを設計するにあたっては、CW練習機として考えられる機能を盛り込むなど、次の8点を目標にしました。

- ① 欧文・和文とも練習可能なこと。
- ② 暗文は、完全にでたらめにするため、BASICの乱数発生機能を使う。
- ③ 暗文の5文字ずつの単位を、何単位にするか指定できるようにする。
- ④ くり返して練習できるよう、リピートの回数を指定できるようにする。
- ⑤ 解答の表示を、モールス符号と同時に、もしくは後から表示するかの指定ができるようにする。
- ⑥ モールス符号の発生は、BASICのMUSIC機能を使う。
- ⑦ 同様にスピードは、TEMPO命令で可変する。
- ⑧ 平文の練習もできるよう、キーボードからの入力も可能にする。

以上の目標は、すべて達成できましたが、BASICのTEMPO命令を使った関係で、スピードを連続的に可変

第1図 プログラムの説明

\*\*\*\*\* ## MAIN ROUTINE ## \*\*\*\*\*      \*\*\*\*\* ## SUB ROUTINE ## \*\*\*\*\*

文番号	プログラムの説明
4	配列の定義
5-17	機械語データの読み込み
40-172	モールス符号データ
500-530	デモ用プログラムおよびスピード、練習方法プログラムを呼び出す
1000-1070	「欧文キーボード入力モード」の処理
2000-2160	「欧文・暗文・数字ありモード」で解答を表示する場合の処理
2500-2720	「欧文・暗文・数字なしモード」で解答をあとから表示する場合の処理
3000-3160	「欧文・暗文・数字なしモード」で解答を同時に表示する場合の処理
3500-3720	「欧文・暗文・数字なしモード」で解答をあとから表示する場合の処理
4000-4070	「和文キーボード入力モード」の処理
5000-5150	「和文・暗文モード」で解答を同時に表示する場合の処理
5500-5720	「和文・暗文モード」で解答をあとから表示する場合の処理
9000-9040	CWトレーナー練習終了時の処理

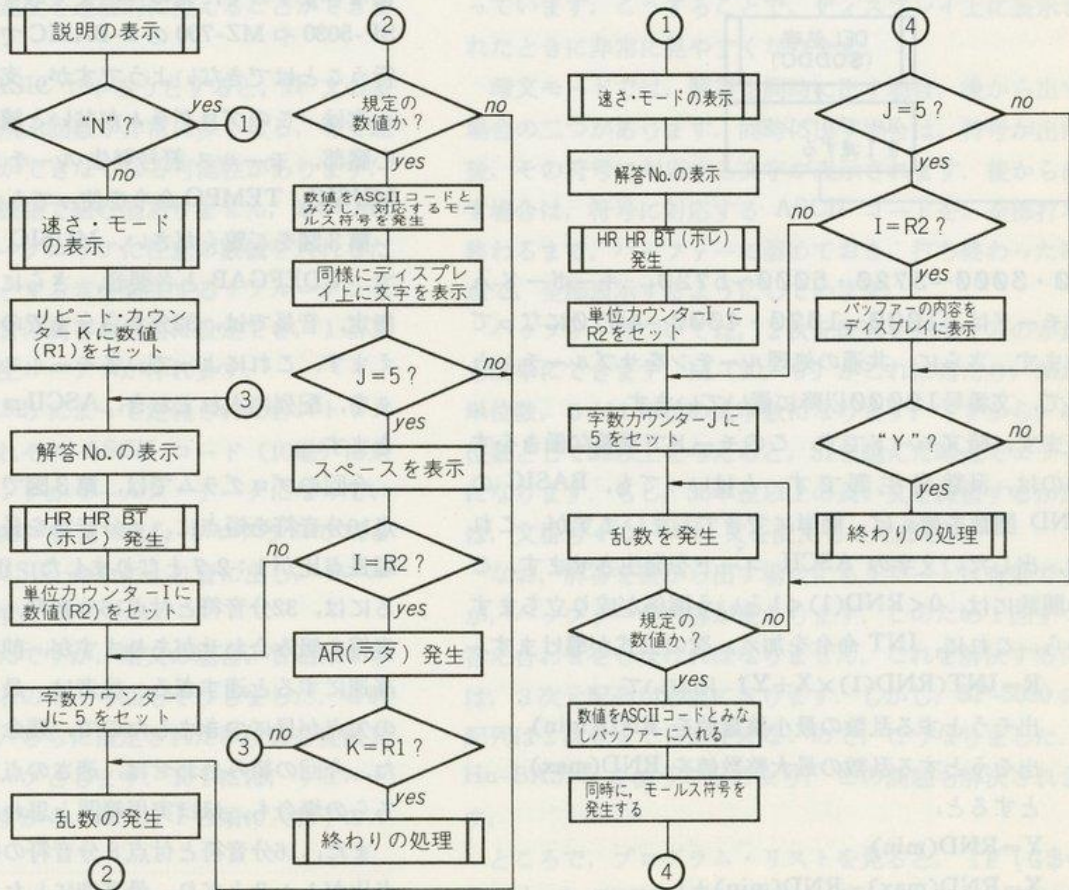
文番号	プログラムの説明
10000-10190	デモプログラム(電けんを表示し,"S"入力待ち)"S"でもどる
10500-10620	練習方法を表示し,"Y"入力待ち,"Y"ならリターン
11000-11130	スピードを表示し,"Y"入力待ち,"Y"ならリターン
12000-12020	練習モードと、速さを表示する
15000-15070	欧・和 キーボード入力モード 時の説明
16000-16110	欧・和・暗文 モード 時の説明
17000-17280	欧・和 キーボード入力モード時の処理
20000-20050	HR HR BT を発生
20100-20150	HR HR ホレ を発生
20200-20220	AR を発生
20300-20320	ラタを発生
25000-25060	各モード終了時の表示

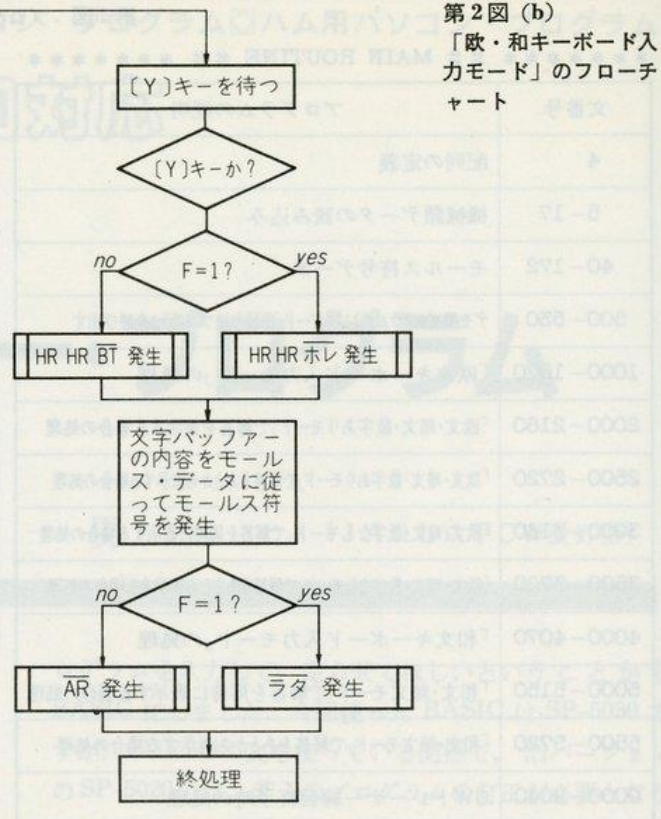
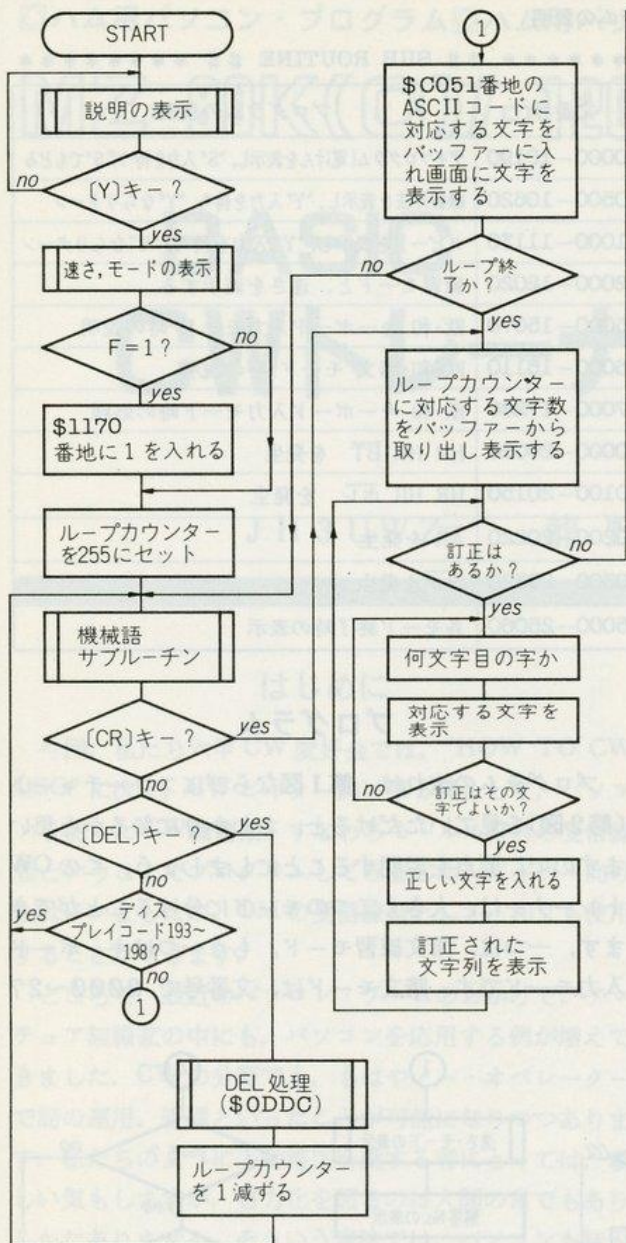
プログラム

プログラムの流れは、第1図ならびにフローチャート(第2図)を見ていただくと、おわかりになるかと思いますが、要点を説明することにしましょう。このCWトレーナーは、大きく二つのモードに分けることができます。一つは、暗文練習モード、もう一つはキーボード入力モードです。暗文モードは、文番号で 2000~27

することができませんでした。スピード・音の高さともに可変にしたがったのですが、これは次の機会にしたいと思います。

第2図 (a)  
「各暗文モード」の  
フローチャート





第2図 (b)  
「暗・和キーボード入力モード」のフローチャート

20・3000~3720・5000~5720, キーボード入力モードは, 1000~1070・4000~4070になっています。さらに, 共通の処理ルーチンをサブルーチンとして, 文番号10000以降に置いています。

まず, 暗文モードです。このモードで重要な働きをするのは, 乱数発生部です。とはいっても, BASIC の RND 関数を使えば, 簡単にできてしまいますが, これで, 出したい文字の ASCII コードを発生させます。この関数には,  $0 < \text{RND}(1) < 1$  という関係が成り立ちますから, これに, INT 命令を加え, 次の公式が導けます。

$$R = \text{INT}(\text{RND}(1) \times X + Y) \text{ において}$$

出そうとする乱数の最小整数値を RND(min)  
 出そうとする乱数の最大整数値を RND(max)  
 とすると,  
 $Y = \text{RND}(\text{min})$   
 $X = \text{RND}(\text{max}) - \text{RND}(\text{min}) + 1$

これから, たとえば, 「暗文・数字ありモード」では, ASCII コードで, 48(0)~90(Z) を出せばよいわけですから,  $R = \text{INT}(\text{RND}(1) \times 43 + 48)$  となることがわかります。このモードでは, ASCII コードで, 58~64は不要になりますので, 後の IF 文で取り除いています。また, SP-5030 や MZ-700 の S-BASIC では, 乱数の初期化を行うことはできないようですが, 支障はないようです。

次は, このプログラムがないと練習機とはならない, 心臓部, モールス符号発生ルーチンです。これは, MUSIC・TEMPO 命令を使ってみました。

第3図をご覧ください。MUSIC 命令では, 音程として, CDEFGAB と各嬰音, さらに2オクターブまでの指定, 音長では, 32分から全までの, 各休符, 音符が使えます。これによって, 各モールス符号に対応するデータを, 配列に入れておき, ASCII コードと一致させておきます。

今回のプログラムでは, 第3図でもわかるように, 付点16分音符を短点に, 4分音符を長点に対応させたので, 短長点比が1:2.7となりました。正確に1:3に合わせるには, 32分音符と付点16分音符, 8分音符と付点4分音符の組み合わせがありますが, 前者は, スピードを最高速にすると速すぎる, 後者は, 最低速で遅すぎるなどの欠点が目につきましたので, 残念ながらあきらめました。今回の組み合わせは, 速さの点で最低速, 最高速どちらの場合も, ほぼ実用範囲と思われる。

また, 16分音符と付点8分音符の組み合わせも, 短長点比が1:3となり, 最高速にしたとき, 少し速すぎ

る点を除けば、けっこう使えるかも知れません。この二者の組み合わせから、速さの点、短長点比などを考慮して、皆さんで選んでください。

音の高さについては、先ほどの音程で述べたように、都合37種類\*もあるわけですから、好きな音の高さにしてください。筆者の考えでは、もう少し高いほうが聞きやすいと思うのですが、ただ、“B”より高くするには、“#B”にするか、“=”(ASCIIコードで215)を付加することになり、キー入力の手間が増えるのでやめました。それでも、もう少し高い音がいいという人は変更してください。

この方式は簡単にモールス符号のデータを作ることができる反面、速度が連続的に可変できません。スピード“6”と“7”で大きな隔りがあります。これを解決するには、符号データから短点、長点の区別を

読み、それぞれのモールス発生ルーチンへ飛ばせばよいことになります。そのとき、短長点比を1：3となるようなディレイ・ルーチンを入れておき、外から指定するようにしてやると、速度を連続的に変えることができます。

ただし、これをBASICでやろうとすると、IF文の多用、サブルーチンへ飛ぶ回数が非常に多くなる、等で速いスピードでの練習ができなくなる可能性があります。こうなると、もう機械語で組む他ありません。幸いMZのモニターには、ワークエリアに任意の数値を入れるだけで、それを分周比とする音を発生するサブルーチンもあることですから、音の高さも自由に設定でき、1石2鳥なモールス符号発生ルーチンが作れます。

さて、配列M\$(192)によって定義された各ストリング変数は、添字がそれぞれASCIIコード(10進)に対応しており、ストリングがモールス・データになっています。いま乱数が(65)を出せば、プログラムでM\$(65)のデータをMUSIC命令により音に出し、モールス符号とするわけです。

次に符号の出しかたですが、暗文の場合、普通は5字が1単位になっているので、今回もそうしました。これを一つのループとし、さらに指定された単位数を変数R2に入れ、単位のループとします。R1には、リピート回数がはいっていますから、K、J、Iの順に入れ子のル

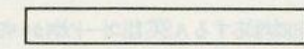
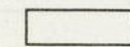
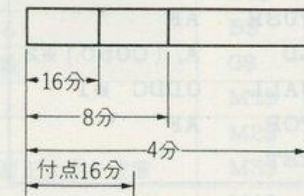
\*異名同音は除く。たとえば#EとFは同じ音

### 第3図 BASICのMUSIC命令の仕組み

休符	♪	♪	♪	♪	♪	♪	♪	♪	♪	♪
符名	32分	16分	付点16分	8分	付点8分	4分	付点4分	2分	付点2分	全
音符										
*	0	1	2	3	4	5	6	7	8	9

\*BASIC上における記号

(1) MUSIC命令で音楽を作る場合、音程(CDEFGABと各嬰音)+上表の音長記号を組み合わせる



#### モールス符号データの作り方

各符の長さの関係は左のようになっている。したがって、これを使ってモールス・データを作るには、休符もうまく組み合わせ、短長点比が1:3になるようにする

このプログラムで作った短点(付点16分音符)

本プログラムで使った長点(4分音符)

この例では、短点・長点比は1:2.7になるが、その理由は本文参照

例 A): = B2R2B5

ア) = B5R2B5R2B2R2B5R2B5

なお、音長を表す数字は、直前の符号と同一である場合は省略してよい

例 A):の場合 B2RB5 ア):の場合 B5R2B5R2BRB5R2B5

ープにすることにより、符号を出します。これについてはあまり問題はないでしょう。なお、ディスプレイにきれいに出すために、単位間には三分のスペースがはいっています。こうすることで、ディスプレイ上に表示されたときに非常に見やすくなります。

暗文モードでは、解答を同時に出す場合、後から出す場合の二つがあります。同時に出す場合は、符号が出た後、その符号に対応する文字が表示されます。後から出す場合は、符号に対応するASCIIコードを、全部打ち終わるまで、バッファに溜めておき、打ち終わった時点で、全部表示するようにしています。

バッファについては、2次元配列を利用するのが最も簡単にできます。M(30, 5)がこれに当たり、30が単位数、5が1単位の文字数になります。ですから、単位数として31以上を与えると、31を越えた時点でエラーになります。もし、30単位以上の長い文を練習するかたは、文番号4の、DIM文を変更してください。

なお、解答を後から出す場合にもリピートは有効ですが、バッファの内容が変わります。このため1回ずつ答え合わせをしなければなりません。これを解決するには、3次元配列が必要になります。しかし、SP-5030の配列は2次元までしか使えないので、こうなりました。Hu-BASICを使うことにより、この問題も解決されます。

ところで、プログラム・リストを見ると、IF(G\$="Y")=0 THEN ~などのわかりにくい条件判断が出

第4図 機械語サブルーチンのプログラム

アドレス (16進)	マシンコード (16進)	マシンコード (10進)	ニーモニックコード
C000	F5	245	PUSH AF
C001	CDB309	205, 179, 9	CALL 09B3 ※1
C004	3250C0	50, 80, 192	LD (C050), A※2
C007	CDCE0B	205, 206, 11	CALL 0BCE ※1
C00A	3251C0	50, 81, 192	LD (C051), A※3
C00D	F1	241	POP AF
C00E	C9	201	RET
C00F	F5	245	PUSH AF
C010	3A50C0	58, 80, 192	LD A, (C050)※2
C013	CDDC0D	205, 220, 13	CALL ODDC ※1
C016	F1	241	POP AF
C017	C9	201	RET

※1 は MZ-80K/C, 1200 システム・モニター(SP-1002)のサブルーチン

MZ-700のシステム・モニター(1Z-009A)も同一アドレス

※2 C050番地には、ディスプレイ・コードがはいる

※3 C051番地には、C050番地のディスプレイ・コードに対応するASCIIコードがはいる

できます。今回のプログラムでは、よく出てきますので少し説明いたします。その前に、ダイレクト・モードで `A$="J":PRINT (A$="H")` を実行してみてください。次に、`A$="J":PRINT (A$="J")` としてみたらどうですか。前者は、0、後者は -1、と表示されるはずで

すなわち、条件判断の結果が“真”なら -1 が、“偽”なら 0 となることがおわかりでしょう。数値の場合は、`IF A<>0 THEN ~` という関係が使えますが、ストリングの場合は、エラーとなりますから `IF G$<>"Y"` として使いたい場合など、非常に困ってしまいます。こんなときには、`IF (G$="Y")=0`、とすれば、まったく同時に使えます。また、シャープの BASIC は、論理演算子、たとえば AND, OR がそれぞれ \* (アスタリスク)、+ と表現されますので、他社の機種に移植されるかたは注意してください。

BASIC の旧バージョンの SP-5020 をお持ちのかたは、本プログラム中の CURSOR 文が使えませんので、CURSOR 文の代わりに、次の文を挿入してください。[`POKE $1171, X:POKE $1172, Y`]\* ここで X, Y は CURSOR x, y と同じ数値です。

暗文モードで重要な点は、だいたい以上です。次にキーボード入力モードに移ります。このモードは、読んで字の如く、キーボードから入力した文字をそのままモジュール符号に打っていきます。最初に制作したプログラムは、このモードに INPUT 文を使っていたため、わずか40字弱しか入れることができず、少し長い文章を入れたいときなど、すぐに一杯になってしまい実用的ではあり

\*\$マークは16進数, X, Y は10進数を表す

ませんでした。

次に考えたのは配列と GET 文を使う方法で、ほぼ実用になる字数を収容することができます。しかし、GET 文を使っているため、[DEL] キーや、[CR] キーなどを使うことができず、不便でした。たとえば、訂正したくてもできませんし、入力終了には、[CR] を使えません。これらのことで苦心して改良したものが、このプログラムです。

BASIC に全面的に依存することをあきらめ、機械語のサブルーチンを、配置し GET 文の代わりに、このサブルーチンを CALL (USR) します。機械語のプログラムは第4図を見ていただければわかるような、わずか24バイトの短いものです。し

かし、このおかげで、キーボード入力モードが大変実用的になりました。このサブルーチンでは、最初に 09B3 番地を CALL していますが、09B3 番地はモニター・サブルーチンで、カーソルをフラッシュしながら1字入力を待ち、キーが押されると、そのキーに対応するディスプレイ・コードを CPU の A レジスターに格納します。

その後、A レジスターのディスプレイ・コードを、C050 番地へ転送し、再びモニター・サブルーチンの、0BCE 番地を CALL すると、ディスプレイ・コードが対応する ASCII コードに変換され、A レジスターにはいるので、今度はこれを C051 番地へ転送し、ひとまず BASIC へもどります。

BASIC 中では、C050 番地にあるディスプレイ・コードを PEEK 文で読みとり、[CR] キーか? [DEL] キーか? それ以外のディスプレイ・コントロール・キーか? 一般のキーか? 以上4種類について判断します。

[CR] であれば入力終了としてループを脱出しますが、[DEL] の場合は、さらに機械語サブルーチンを CALL することになります。この場合は、C00F 番地を CALL するわけですが、C00F 番地からの処理は、A レジスターにディスプレイ・コードを入れてから、モニター・サブルーチンの 0DDC 番地を呼び出し、A レジスターのディスプレイ・コードに該当するディスプレイ・コントロールを行います。これで、[DEL], [CR] が有効に使えるようになります。

その他のディスプレイ・コントロール・キーは受け付けませんが、もし必要なら訂正してください。なお、和文入力モードでは、[カナ] キーを押す必要はありません。この処理は、文番号 17010 で行っており、モニターのワークエリアである 1170 番地に、1 を書き込むこと

第5図 プログラム中で使っている変数およびストリング

変数名	用 途	配 列	用 途
A	\$C051番地のASCIIコードをPEEKで読み出し入れる	C\$(255)	「キーボード入力モード」時のASCIIコード・バッファ
F	欧・和「キーボード入力モード」時に0が、和なら1がある	M\$(192)	モールスコード・データ
G	GET文で受け取った数値	M(30,5)	「暗文モード」時、解答をあとから表示するためのバッファ
I	ループ変数		
J	ループ変数	ストリング	用 途
K	ループ変数	A1\$	“*** CW TRAINER ***”
M	READ・DATAで読み込んだ、マシン語データ	A2\$	“1983年3月21日 PROGRAMED BY JH3UWZ”
N	スピード	A3\$	“ハジメルトキハ〔S〕ヲ オシテクダサイ”
R	欧・和「暗文モード」で乱数値を入れる	B\$	各モード名 例“ワブン キーボード ニュウリョク”
R1	欧・和「暗文モード」でリピートの回数	G\$	GET文で受け取ったストリング
R2	欧・和「暗文モード」で単位数	M1\$	“HR HR =” (HR HR BT)
S	練習方法選択時の選択した数値	M2\$	“HR HR #” (HR HR ホレ)
T	欧・和「キーボード入力モード」時の訂正文字の位置	M3\$	“+” (AR)
		M4\$	“\$” (ラタ)

でキーボードをカナ・モードにしています。

また、1170番地に0を書き込めば、キーボードは英数モードになりますが、このプログラム中では、これらのごとをすべてプログラム中で処理していますので、とくにキー入力モードでは「カナ・英数キー」の存在を気にしなくてもよいようになっています。

ところで、キーボード入力モードでの入力文字数は、255字\*以内に制限されます。これは、文字バッファに配列、C\$(255)を使ったためです。Hu-BASICでは、この要素の制限がないので、もう少し増やすこともできますが、もっともこれだけあれば、平文の練習であまり不便を感じることはないと思われます。

入力途中での訂正は、[DEL]キーを使えばいいのですが、いったん[CR]を押してしまうと、訂正はできなくなりますので、その対策として、[CR]後の訂正に少し凝ってみました。とはいっても、何文字目を訂正するかを指定し、その文字を訂正する文字と交換するだけの簡単なものです。それでも、平文を打つ場合には、案外便利なもので、最初に作った練習プログラムには、一切訂正機能がなかったことを考えると、ずいぶん楽になりました。

このモードのモールス発生ルーチンについては、暗文モードとまったく同じなのですが、和文モードと欧文モードが共通のサブルーチンのため、先ほど述べた変数(F)の値に応じて、[HR HR BT]を出すか、[HR HR ホレ]を出すかの判断をする部分がいります。

各モード処理ルーチンの最後には、終処理があり、①もう一度 ②スピード変更 ③練習方法変更 ④練習終了の中から一つを選択します。④の場合は、文番号9000番に飛んで、CW トレーナー・プログラムを終

\*本当は配列は0から有効なので256字であるが、プログラムのわかりやすさから255字にした。

了することになります。以上がプログラムの要点です。なお、プログラム中で使っている変数およびストリングを第5図に示します。

### CW トレーナーの使い方

まずプログラムをどンドン打ち込んでください。プログラムがはいっていないことには話になりません。文番号15~17の機械データは、数字が一つ間違っても暴走のもとですから注意してください。プログラムがはいったらとりあえず、カセットにSAVEしておくと、もし暴走してしまったときなど、再びテープをLOADして、デバックできます。

では、プログラムのはいったところで、RUNさせます。しばらく沈黙していますが、この間に機械語の書き込み、モールス・データの読み込みをしています。これらデータの読み込みが終わったところで、電けんがディスプレイされ、“S”キーを待ちます。“S”が押されると次に練習方法、スピードの順に表示され、それぞれ希望のところまで“Y”キーを押してください。

なお、スピードのところでは、1分間に打たれる文字数が表示されますが、(例:[1]では15LTRS/分)これは、あくまでもだいたいの目安にしてください。MZ内部のTEMPO調整のVRでも大きく違ってきます。

スピードの指定まで終了すると、いよいよそれぞれの処理ルーチンへ飛ぶわけですが、まず暗文モードを指定した場合、リピート回数を聞いてきます。リピート回数というのは、[HR HR BT ~本文~ AR]を何回くり返すかの意味です。次は単位数を聞いてきます。暗文とは、普通5字を1単位として構成されていますから、その5字ずつの組み合わせを、1単位として何単位にするかの意味です。

暗文モードで最後に聞いてくるのは、解答を同時に出すか後から出すかの指定です。これについては、プログ





```

4 DIM C$(255),M$(192),M(30,5)
5 LIMIT#BFFF:FOR I=49152TO49175:READ M:POKE I,M:NEXT I
10 PRINT "G":CURSOR 4,12:PRINT "DATA ヲ モンテイマス。 シンラウオマチクワサイ !! "
15 DATA 245,205,179,9,50,80,192,205,206
16 DATA 11,50,81,192,241,201,245,58,80
17 DATA 192,205,220,13,241,201
40 REM      [ コウウ "(" - "/" ]
43 M$(32)="R5"
44 M$(35)="B5R2BRBRB5R2B5R2B5":REM トレ #
45 M$(36)="B2RBRBRB5R2B"
46 M$(38)="B2R5B2RBRB"
47 M$(40)="B5R2BRB5R2B5R2BRB5"
48 M$(41)="B2RB5R2BRBRB5R2B"
49 M$(42)="B5R2BRBRB5"
50 M$(43)="B2RB5R2BRB5R2B"
51 M$(44)="B5R2B5R2BRBRB5R2B5"
52 M$(45)="B5R2BRBRBRBRB5"
53 M$(46)="B2RB5R2BRB5R2BRB5"
54 M$(47)="B5R2BRBRB5R2B"
60 REM      [ スラシ ]
61 M$(48)="B5R2B5R2B5R2B5R2B5"
62 M$(49)="B2RB5R2B5R2B5R2B5"
63 M$(50)="B2RB2RB5R2B5R2B5"
64 M$(51)="B2RBRBRB5R2B5"
65 M$(52)="B2RBRBRBRB5"
66 M$(53)="B2RBRBRBRB"
67 M$(54)="B5R2BRBRBRB"
68 M$(55)="B5R2B5R2BRBRB"
69 M$(56)="B5R2B5R2B5R2BRB"
70 M$(57)="B5R2B5R2B5R2B5R2B"
71 M$(61)="B5R2BRBRBRB5"
72 M$(63)="B2RBRB5R2B5R2BRB"
75 REM      [ アルファベット ]
76 M$(65)="B2RB5"
77 M$(66)="B5R2BRBRB"
78 M$(67)="B5R2BRB5R2B"
79 M$(68)="B5R2BRB"
80 M$(69)="B2"
81 M$(70)="B2RBRB5R2B"
82 M$(71)="B5R2B5R2B"
83 M$(72)="B2RBRBRB"
84 M$(73)="B2RB"
85 M$(74)="B2RB5R2B5R2B5"
86 M$(75)="B5R2BRB5"
87 M$(76)="B2RB5R2BRBR"
88 M$(77)="B5R2B5"
89 M$(78)="B5R2B"
90 M$(79)="B5R2B5R2B5"
91 M$(80)="B2RB5R2B5R2B"
92 M$(81)="B5R2B5R2BRB5"
93 M$(82)="B2RB5R2B"
94 M$(83)="B2RBRB"
95 M$(84)="B5"
96 M$(85)="B2RBRB5"
97 M$(86)="B2RBRBRB5"
98 M$(87)="B2RB5R2B5"
99 M$(88)=M$(42)
100 M$(89)="B5R2BRB5R2B5"
101 M$(90)="B5R2B5R2BRB"
110 REM      [ ワフン ]
111 M$(129)=M$(32)
112 M$(131)="B2RB5R2BRB5R2BRB"
113 M$(132)="B2RB5R2BRB5R2BRB5"
114 M$(134)=M$(74)
115 M$(135)="B5R2B5R2BRB5R2B5"
116 M$(136)=M$(65)
117 M$(137)=M$(85)
118 M$(138)="B5R2BRB5R2B5R2B5"
119 M$(139)="B2RB5R2BRBRB"
120 M$(140)=M$(87)
121 M$(141)="B5R2BRBRB5R2B5"
122 M$(142)=M$(77)
123 M$(143)=M$(80)
124 M$(144)="B2RB5R2B5R2BRB5"
125 M$(145)="B5R2B5R2BRB5R2B5"
126 M$(146)=M$(65)
127 M$(147)=M$(85)
128 M$(148)="B5R2BRB5R2B5R2B5"
129 M$(149)="B2RB5R2BRBRB"
130 M$(150)=M$(76)
131 M$(151)="B5R2BRB5R2BRB"
132 M$(152)=M$(86)
133 M$(153)=M$(89)
134 M$(154)="B5R2B5R2B5R2B5"
135 M$(155)="B5R2BRB5R2BRB5"
136 M$(156)="B5R2B5R2BRB5R2B"
137 M$(157)="B5R2B5R2B5R2BRB5"
138 M$(158)="B2RB5R2B5R2B5R2B"
139 M$(159)="B5R2B5R2B5R2B"
140 M$(160)=M$(78)
141 M$(161)=M$(70)
142 M$(162)=M$(80)
143 M$(163)="B2RB5R2BRB5R2B5"
144 M$(164)="B2RBRB5R2BRB"
145 M$(165)=M$(82)
146 M$(166)=M$(67)
147 M$(167)=M$(72)
148 M$(168)=M$(81)
149 M$(169)="B2RBRB5R2B5"
150 M$(170)=M$(66)
151 M$(171)="B5R2B5R2BRBRB5"
152 M$(172)=M$(90)
153 M$(173)=M$(69)
154 M$(174)=M$(68)
155 M$(175)=M$(42)
156 M$(176)="B2RBRB5R2BRB5"
157 M$(177)=M$(84)
158 M$(178)=M$(61)
159 M$(179)="B5R2BRBRB5R2B"
160 M$(180)=M$(87)
161 M$(181)="B5R2BRBRB5R2B5"
162 M$(182)=M$(77)
163 M$(183)=M$(83)
164 M$(184)=M$(71)
165 M$(185)="B5R2BRB5R2B5R2B"
166 M$(186)=M$(79)
167 M$(187)="B2RB5R2BRB5"
168 M$(188)=M$(75)
169 M$(189)=M$(43)
170 M$(190)=M$(73)
171 M$(191)="B2RBRB5R2B5R2B"
172 M$(192)=M$(32)

```

```

175 REM *****
176 REM      [ MAIN ROUTINE ]
177 REM *****
180 A1$="      ** CW  TRAINER **      "
181 A2$=" 1983年 3月 21日      PROGRAMED BY JH3UWZ "
182 A3$=" シンラウオマチクワサイ !! [S] ヲ オ シ テ ク ワ サ イ !! "

```

```

190 M1#="HRHR="
191 M2#="HRHR#"
192 M3#="+ "
193 M4#="# "
500 GOSUB 10000
510 GOSUB 10500
520 GOSUB 11000
530 ON S GOTO1000,2000,3000,4000,5000
1000 B#="オウフン キ-ホ-ト` ニュウリョク":F=0
1010 GOSUB 15000
1020 GOSUB 12000
1030 GOSUB 17000
1040 GOSUB 18000
1050 GOSUB 25000
1060 ON G GOTO1020,1070,510,9000
1070 GOSUB 11000:GOTO 1020
2000 B#="オウフン アンフン スウシ`アリ"
2010 GOSUB 16000:IF G#="N" THEN 2500
2020 FOR I=1TO500:NEXT:GOSUB 12000
2030 FOR K=1TOR1
2040 PRINT "SS":PRINT "[ カイトウ No. ";K;" ]":PRINT
2050 GOSUB 20000
2060 FOR I=1TOR2:FOR J=1TO5
2070 R=INT(RND(1)*43+48):IF ((R<48)+(R>57))*((R<65)+(R>90)) THEN 2070
2080 MUSIC M$(R):"R5":PRINT CHR$(R);
2090 NEXT J
2100 MUSIC "R7":PRINT " ";
2110 NEXT I
2120 GOSUB 20200
2130 MUSIC "R8R":NEXT K
2140 PRINT "SSSSSSSS":GOSUB 25000
2150 ON G GOTO2020,2160,510,9000
2160 GOSUB 11000:GOTO 2020
2500 FOR I=1TO500:NEXT I:GOSUB 12000
2510 FOR K=1TOR1
2520 CURSOR 0,22:PRINT " * カイトウ、アトカラ ヒョウシ`サレマス。 * "
2530 CURSOR 4,9:PRINT "[ カイトウ No. ";K;" ]S"
2540 GOSUB 20000
2550 FOR I=1TOR2:FOR J=1TO5
2560 R=INT(RND(1)*43+48):IF ((R<48)+(R>57))*((R<65)+(R>90)) THEN 2560
2570 M(I,J)=R:MUSIC M$(R):"R5"
2580 NEXT J
2590 MUSIC M$(32)
2600 NEXT I
2610 GOSUB 20200
2620 FOR I=1TO500:NEXT
2630 CURSOR 0,11:FOR I=1TOR2:FOR J=1TO5
2640 PRINT CHR$(M(I,J));
2650 NEXT J:PRINT " ";:NEXT I
2660 CURSOR 3,22:PRINT "* コノイ アツカ`オホシ` [Y]オオシ`ク`サイ * "
2670 GET G#:IF (G#="Y")=0 THEN 2670
2680 CURSOR 0,11:FOR I=1TO12:PRINT SPC(40)::NEXT I
2690 NEXT K
2700 GOSUB 25000
2710 ON G GOTO 2500,2720,510,9000
2720 GOSUB 11000:GOTO 2500
3000 B#="オウフン アンフン スウシ`ナシ"
3010 GOSUB 16000:IF G#="N" THEN 3500
3020 FOR I=1TO500:NEXT:GOSUB 12000
3030 FOR K=1TOR1
3040 PRINT "SS":PRINT "[ カイトウ No. ";K;" ]":PRINT
3050 GOSUB 20000
3060 FOR I=1TOR2:FOR J=1TO5
3070 R=INT(RND(1)*26+65)
3080 MUSIC M$(R):"R5":PRINT CHR$(R);
3090 NEXT J
3100 MUSIC "R7":PRINT " ";
3110 NEXT I
3120 GOSUB 20200
3130 MUSIC "R8R":NEXT K
3140 PRINT "SSSSSSSS":GOSUB 25000
3150 ON G GOTO 3020,3160,510,9000

```

```

3160 GOSUB 11000:GOTO 3020
3500 FOR I=1T0500:NEXT I:GOSUB 12000
3510 FOR K=1TOR1
3520 CURSOR 0,22:PRINT "      * カイトウ、アトカラ ヒョウシガサレマス。 *      "
3530 CURSOR 4,9:PRINT "[ カイトウ No.  ";K;" ]0"
3540 GOSUB 20000
3550 FOR I=1TOR2:FOR J=1T05
3560 R=INT(RND(1)*26+65)
3570 M(I,J)=R:MUSIC M$(R);"R5"
3580 NEXT J
3590 MUSIC M$(32)
3600 NEXT I
3610 GOSUB 20200
3620 FOR I=1T0500:NEXT
3630 CURSOR 0,11:FOR I=1TOR2:FOR J=1T05
3640 PRINT CHR$(M(I,J));
3650 NEXT J:PRINT"      ";:NEXT I
3660 CURSOR 3,22:PRINT "* コソイ アツカオ オケルン [Y]ヲ オシテクダサイ *"
3670 GET G$:IF (G$="Y")=0 THEN 3670
3680 CURSOR 0,11:FOR I=1T012:PRINT SPC(40));:NEXT I
3690 NEXT K
3700 GOSUB 25000
3710 ON G GOTO 3500,3720,510,9000
3720 GOSUB 11000:GOTO3500
4000 B$="ワフン キ-ホ-ト ニュウヨク":F=1
4010 GOSUB 15000
4020 GOSUB 12000
4030 GOSUB 17000
4040 GOSUB 18000
4050 GOSUB 25000
4060 ON G GOTO 4020,4070,510,9000
4070 GOSUB 11000:GOTO 4020
5000 B$="ワフン アンフン スウジナシ"
5010 GOSUB 16000:IF G$="N" THEN 5500
5020 FOR I=1T0500:NEXT:GOSUB 12000
5030 FOR K=1TOR1
5040 PRINT "00":PRINT "[ カイトウ No.  ";K;" ]":PRINT
5050 GOSUB 20100
5060 FOR I=1TOR2:FOR J=1T05
5070 R=INT(RND(1)*58+134)
5080 MUSIC M$(R);"R5":PRINT CHR$(R);
5090 NEXT J
5100 MUSIC "R7":PRINT "      ";
5110 NEXT I
5120 GOSUB 20300
5130 MUSIC "R8":NEXT K
5140 PRINT "00000000":GOSUB 25000
5150 ON G GOTO 5020,5160,510,9000
5160 GOSUB 11000:GOTO5020
5500 FOR I=1T0500:NEXT I:GOSUB 12000
5510 FOR K=1TOR1
5520 CURSOR 0,22:PRINT "      * カイトウ、アトカラ ヒョウシガサレマス。 *      "
5530 CURSOR 4,9:PRINT "[ カイトウ No.  ";K;" ]0"
5540 GOSUB 20100
5550 FOR I=1TOR2:FOR J=1T05
5560 R=INT(RND(1)*58+134)
5570 M(I,J)=R:MUSIC M$(R);"R5"
5580 NEXT J
5590 MUSIC M$(32)
5600 NEXT I
5610 GOSUB 20300
5620 FOR I=1T0500:NEXT
5630 CURSOR 0,11:FOR I=1TOR2:FOR J=1T05
5640 PRINT CHR$(M(I,J));
5650 NEXT J:PRINT"      ";:NEXT I
5660 CURSOR 3,22:PRINT "* コソイ アツカオ オケルン [Y]ヲ オシテクダサイ *"
5670 GET G$:IF (G$="Y")=0 THEN 5670
5680 CURSOR 0,11:FOR I=1T012:PRINT SPC(40));:NEXT I
5690 NEXT K
5700 GOSUB 25000
5710 ON G GOTO 5500,5720,510,9000
5720 GOSUB11000:GOTO5500
9000 PRINT "0":FOR I=1T01000:NEXT
9010 PRINT A1$

```

```

9020 CURSOR 14,16:PRINT "7 3 / 8 8 !!"
9030 TEMPO 7:MUSIC M$(55):"R5":M$(51):"R5B2R5B2"
9040 LIMITMAX:CURSOR0,23:END
10000 REM *****
10001 REM [SUBROUTINE]
10002 REM *****
10010 PRINT "G":CURSOR 5,6:PRINT "███"
10020 CURSOR 6,7:PRINT "■"
10030 CURSOR 6,8:PRINT "■"
10040 CURSOR 6,9:PRINT "███"
10050 CURSOR 6,10:PRINT "███"
10060 CURSOR 6,11:PRINT "███"
10070 CURSOR 5,12:PRINT "██████████████████████"
10080 CURSOR 6,13:PRINT "███"
10090 CURSOR 6,14:PRINT "███"
10100 CURSOR 6,15:PRINT "███"
10110 CURSOR 4,16:PRINT "██████████████████████"
10120 CURSOR 4,17:PRINT "██████████████████████"
10130 CURSOR 4,18:PRINT "██████████████████████"
10140 FOR I=0T039:CURSOR I,1:PRINT MID$(A1$,I,1):
10150 CURSOR I,22:PRINTMID$(A2$,I,1):USR($3E):NEXT I:FOR J=1T0500:NEXT
10160 CURSOR0,22:FOR I=1TOLEN(A3$):FORJ=1T040:NEXT:GET G$:IF G$="S"THEN RET RM
10170 PRINTMID$(A3$,I,1):NEXT I:FORI=1T0500:NEXT
10180 CURSOR0,22:FOR I=1TOLEN(A2$):FORJ=1T075:NEXT:PRINTMID$(A2$,I,1):NEXT
10190 FOR I=1T050:GET G$:IF G$="S" THEN RETURN
10200 NEXT:GOTO 10160
10500 PRINT "G":PRINT A1$
10510 CURSOR 7,4:PRINT "* レンジュウ入シカタ イラテクダサイ *"
10520 CURSOR 6,8:PRINT "[1] -- オフン (キ-ホ-ト) ニウリョク )"
10530 CURSOR 6,10:PRINT "[2] -- オフン (アフン スジアリ )"
10540 CURSOR 6,12:PRINT "[3] -- オフン (アフン スジナシ )"
10550 CURSOR 6,14:PRINT "[4] -- クワン (キ-ホ-ト) ニウリョク)"
10560 CURSOR 6,16:PRINT "[5] -- クワン (アフン スジナシ )"
10570 CURSOR 5,20:PRINT "* キホウリトコテ、[Y]ヲ、オシテクダサイ !!*"
10580 FOR I=1T05:CURSOR 7,6+2*I:PRINT ">":USR($3E)
10590 FOR J=1T020:GET G$:IF G$="Y" THEN 10620
10600 NEXT J:CURSOR 7,6+2*I:PRINT STR$(I):NEXT I
10610 GOTO 10580
10620 S=I:RETURN
11000 PRINT "G":PRINT A1$
11010 CURSOR 4,5:PRINT "* スピ-トヨ イラテ [Y]ヲ オシテクダサイ *"
11020 CURSOR 5,8:PRINT "[1] -- タイン オイ ( 15 LTRS/分)"
11030 CURSOR 5,10:PRINT "[2] -- スコシ オイ ( 20 LTRS/分)"
11040 CURSOR 5,12:PRINT "[3] -- オオイ ( 25 LTRS/分)"
11050 CURSOR 5,14:PRINT "[4] -- フツク ( 30 LTRS/分)"
11060 CURSOR 5,16:PRINT "[5] -- ハヤイ ( 40 LTRS/分)"
11070 CURSOR 5,18:PRINT "[6] -- スコシ ハヤイ ( 65 LTRS/分)"
11080 CURSOR 5,20:PRINT "[7] -- タイン ハヤイ (115 LTRS/分)"
11090 FOR I=1T07:CURSOR 6,6+2*I:PRINT ">":USR($3E)
11100 FOR J=1T020:GET G$:IF G$="Y" THEN 11130
11110 NEXT J:CURSOR 6,6+2*I:PRINT STR$(I):NEXT I
11120 GOTO 11090
11130 N=I:RETURN
12000 PRINT "G":PRINT A1$
12010 CURSOR 0,3:PRINT "[ ェ - ト ] ";B$:CURSOR 0,5:PRINT "[ ハ ナ ナ ]",N
12020 RETURN
15000 PRINT "G":PRINT A1$
15010 CURSOR 9,3:PRINT "キ-ホ-トカ、モシヨ イレテクダサイ。"
15020 CURSOR 9,5:PRINT "タダシ モシハ、255ジイナイ テス。"
15030 CURSOR 9,7:PRINT "テイセイコト、[DEL] キ-ヲツカイ"
15040 CURSOR 9,9:PRINT "ニウリョクカ、オレヨ、[CR] キ-ヲ"
15050 CURSOR 9,11:PRINT "オシテクダサイ。[カナ] キ-ハ、オナ"
15060 CURSOR9,13:PRINT "クモ カマイセン。テ、シユシカ"
15070 CURSOR9,15:PRINT "テ、カラ、[Y] キ-ヲ、オシテクダサイ !"
15080 GET G$:IF G$="Y" THEN RETURN
15090 GOTO 15080
15100 GET G$:IF G$="Y" THEN RETURN
16000 PRINT "G":PRINT A1$
16010 CURSOR 8,3:PRINT "マイコンカ、ランダム CWヲ ウチマシテ、"
16020 CURSOR 8,5:PRINT "シユシシテクダサイ。マス、リヒ-ト カイ"
16030 CURSOR 8,7:PRINT "スウカ、イレテクダサイ。"
16040 CURSOR 22,7:INPUT R1
16050 CURSOR 22,5:PRINT "ツキニ、タイン ナン"

```

```

16060 CURSOR 8,7:PRINT "カニ シマスカ":SPC(10)
16070 CURSOR 17,7:INPUT R2
16080 CURSOR 22,5:PRINT "カトウの トウシ"
16090 CURSOR 8,7:PRINT "タシマスカ ? [Y] OR [N] "
16100 GET G#: IF (G#="Y")+(G#="N")=0 THEN 16100
16110 RETURN
17000 CURSOR 0,22:PRINT "* テイシの [DEL]キ、オウシの [CR]キ、オシテクタサイ *"
17010 IF F=1 THEN POKE#1170,1
17020 CURSOR 0,9:PRINT "> ":FOR I=1TO255
17030 USR(#C000):A=PEEK(#C050):IF A=205 THEN 17150
17040 IF (A=199)+(A=200) THEN 17080
17050 IF (A>192)*(A<203) THEN GOTO 17030
17060 IF A=0 THEN 17100
17070 GOTO 17110
17080 USR(#C00F)
17090 I=I-1:GOTO 17030
17100 C#(I)=" ":USR(#3E):PRINT " ":GOTO 17140
17110 G#=CHR$(PEEK(#C051))
17120 USR(#3E)
17130 C#(I)=G#:PRINT G#
17140 NEXT I
17150 CURSOR 0,9:PRINT "> ":FOR J=1TOI-1:PRINT C#(J):NEXT:PRINT " <":K=I-1
17160 PRINT:PRINT " シマスカ=":K:CURSOR 0,22:PRINT SPC(40)
17170 CURSOR 5,22:PRINT "テイシの トウシ、 シマスカ ? [Y] OR [N]"
17180 POKE #1170,0:GET G#:IF G#="N" THEN RETURN
17190 IF (G#="Y")=0 THEN 17180
17200 CURSOR 16,22:PRINT "カニシマスカ ?"
17210 CURSOR 30,22:INPUT T
17220 CURSOR 11,22:PRINT "テイシの [";C#(T);"] シマスカ ? [Y] OR [N] "
17230 GET G#:IF G#="N" THEN 17200
17240 IF (G#="Y")=0 THEN 17230
17250 IF F=1 THEN POKE#1170,1
17260 CURSOR 11,22:PRINT "テイシの シマスカ !!":SPC(12)
17270 CURSOR 29,22:INPUT G#
17280 C#(T)=G#:POKE#1170,0:GOTO 17150
18000 CURSOR 1,22:PRINT "* シュウカ、 シマスカ [Y]ヲ オシテクタサイ !! * "
18010 GET G#:IF (G#="Y")=0 THEN 18010
18020 CURSOR 0,22:PRINT SPC(40)
18030 IF F=1 THEN GOSUB 20100:GOTO 18050
18040 GOSUB 20000
18050 FOR I=1TOK:J=ASC(C#(I)):MUSIC M#(J):"R5":NEXT
18060 IF F=1 THEN GOSUB 20300:RETURN
18070 GOSUB 20200
18080 RETURN
20000 TEMPO N
20010 FOR I=1TO LEN(M1#)
20020 J=ASC(MID$(M1#,I,1))
20030 MUSIC M#(J):"R5"
20040 NEXT I
20050 RETURN
20100 TEMPO N
20110 FOR I=1TO LEN(M2#)
20120 J=ASC(MID$(M2#,I,1))
20130 MUSIC M#(J):"R5"
20140 NEXT I
20150 RETURN
20200 TEMPO N
20210 MUSIC M#(43)
20220 RETURN
20300 TEMPO N
20310 MUSIC M#(36)
20320 RETURN
25000 CURSOR 6,20:PRINT "* オナシのホウキの モウイキト -- [1] *"
25010 CURSOR 6,21:PRINT "* スピードの カイル -- [2] *"
25020 CURSOR 6,22:PRINT "* レンジュウのホウキの カイル -- [3] *"
25030 CURSOR 6,23:PRINT "* レンジュウの ホール -- [4] *"
25040 CURSOR 11,18:PRINT "* シマスカ エラシテクタサイ *"
25050 GET G:IF (G<1)+(G>4) THEN 25040
25060 RETURN

```