

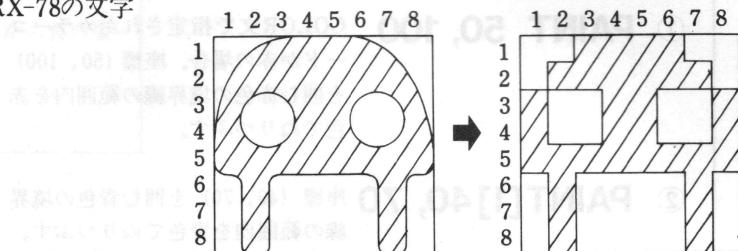
君が作るアニメキャラクター PATTERN(2)

[アニメをストリングデータにする方法]

- ①アニメを方眼用紙に描く。この際、1つのます目が、RX-78の1ドットになるので、大きさを考えること。(6×8がRX-78の文字やカーソルの大きさと同じである)
- ②アニメをドットパターンに直す。

(例)

- ①方眼用紙にアニメを描く
- ②アニメをドットパターンに直す



- ③ドットパターンを左より8つずつに区切り、表1を見ながら右4つ、左4つをそれぞれコードに直す。

- ③表1を見てドットパターンをコードに直す

- ④コードを数字(アスキイ-10進コード)に直す。

よこの列 (上位ビット)	たての列 (下位ビット)	アスキイ-10進コード
□	□	3 C
□	□	7 E
□	□	9 9
□	□	9 9
□	□	F F
□	□	4 2
□	□	4 2
□	□	4 2

- ④コードをアスキーコード表(158ページ)を見ながら、左側に書いてある数字(アスキイ-10進コード)に直す。

- ⑤PATTERN文に与える(リストを参照)

表1

コード	ドットパターン
0	□□□□□□
1	■□□□□□
2	□■□□□□
3	□□■□□□
4	□□□■□□
5	□□□□■□
6	□□□□□■
7	■■■■■■
8	□□□□□□
9	■□□□□□
A	□■□□□□
B	□□■□□□
C	□□□■□□
D	□□□□■□
E	□□□□□■
F	■■■■■■

- ⑤数字をCHR\$で読み込み、PATTERN文に与える。
表の場合ではA\$=CHR\$(60)+CHR\$(126)+...+CHR\$(66)

(注) アスキイ-10進コードに直さず、直接文字や記号のストリングデータで与えてもよい。

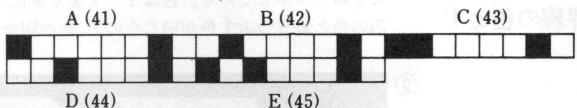
列数

- ドットパターンを何列で描くかを決める。列数分だけ右へドットパターンを描いていきその列数を超えると、その下へ描いていく。

- 列数の制限は、1~24までである。

(例)

```
10 A$ = "ABCDE"
20 PATTERN 3, A$
```



色をいろいろぬり分ける

ペイント PAINT

使用例

① PAINT 50, 100

COLOR文で指定されたカラーコードが赤の場合、座標(50, 100)を囲む赤色の境界線の範囲内を赤色でぬりつぶす。

② PAINT[1]40, 70

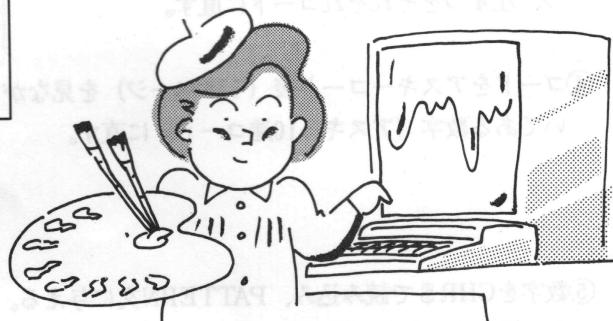
座標(40, 70)を囲む青色の境界線の範囲内を青色でぬりつぶす。

③ PAINT[2]100, 80, 1, 4, 5

座標(100, 80)を含む青、緑および水色の境界線によって囲まれた範囲内を赤でぬりつぶす。

一般型式

PAINT [カラーコード] X座標、Y座標、境界線の色、境界線の色、……



説明

●指定した座標の点を囲む、指定した色の境界線の範囲

内をカラーコードの色でぬりつぶす。

●カラーコードはCOLOR文(89ページ)と同じで、0

～7の数字で指定する。

指定しない場合は、前のCOLOR文で指定したカラーコードの色となる。

●X、Y座標は、色のぬりつぶしを始める点を指定する。

ぬりつぶしたい範囲内の点であれば、どの点を指定してもよい。

X座標は0～191、Y座標は0～183である。

●境界線の色はぬりつぶしたい範囲の境界線の色をカラーコードで指定する。境界が複数の色で囲まれている場合は、そのすべてを指定しなければならない。

指定しない場合は、ぬりつぶしの色を境界線の色とする。

①

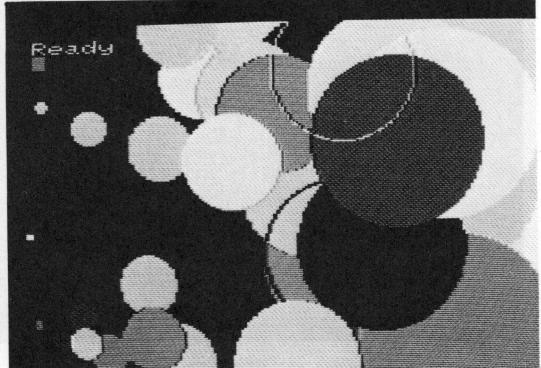
```

LIST REM ** PAINT サンプル ** COLOR ,0 :C
=0 FOR I=1 TO 50
40 X=INT(RND(1)*199)
50 Y=INT(RND(1)*189)
60 R=INT(823)
70 CIRCLE(I,I) X,Y,R
80 PAINT(C) X,Y
90 C=C+1
100 IF C>? THEN C=1
110 NEXT I
120 END
Ready

```

行番号70のCIRCLE文で描いた円の内側を、行番号80のPAINT文でぬりつぶしている。色は1～7まで変化させている。円の色とぬりつぶす色が同じなので、色が円からはみ出さない。

②



注1. PAINT命令を実行してもぬりつぶしを行なわない場合は、再度X座標、Y座標指定しなおしてください。

注2. 複雑な形のぬりつぶしを行なった時Error 6が表示される場合があります。再度X座標、Y座標を指定しなおしてください。

色の引き算

シー エル エス
CLS

使用例

CLS 1

現在表示されている色から青の要素を消す。

一般型式 **CLS** カラーコード

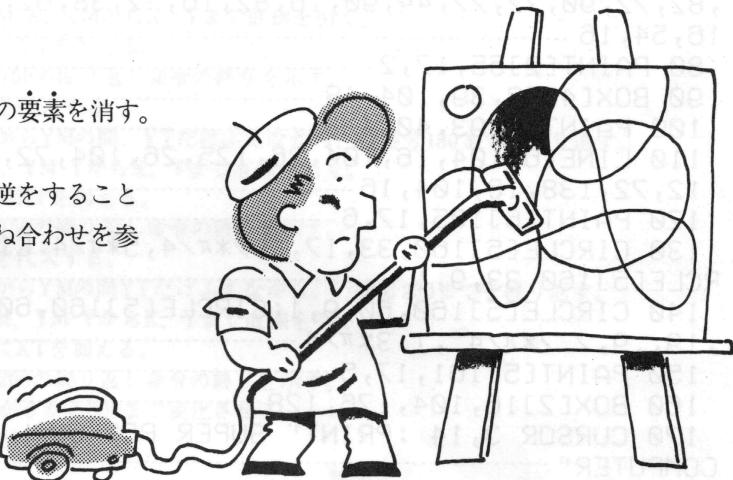
説明 ●現在表示されている色から、指定した色の要素を消す。

ただし、バックカラーの色はかわらない。

●色の要素を消すとは、色の重ね合わせの逆をすることであるので、SET (91ページ) の色の重ね合わせを参考にしよう。

(例)

- ①青から赤を消す→かわらない
- ②紫から赤を消す→青
- ③黄から緑を消す→赤
- ④黄から赤を消す→緑
- ⑤黄から青を消す→かわらない
- ⑥黄から水色を消す→黄から緑と青を消す→赤
- ⑦紫から黄を消す→紫から緑と赤を消す→青



```
① 10 REM ** CLSサンプル ***
20 COLOR ,0
30 FOR CC=0 TO 7
40 PRINT CHR$(6); "-- CLS"; CC;
"__"
50 Y=0 : C=0
60 FOR I=1 TO 2
70 V=Y+70
80 X=-10
90 FOR J=1 TO 4
100 FOR K=0 TO 1000 :NEXT K
110 X=X+40
120 CIRCLE C,X,Y,20
130 PAINT C,X,Y,7
140 C=C+1
150 NEXT J,I
160 FOR L=0 TO 1000 :NEXT L
170 CLS CC
180 FOR CC=L TO 1000 :NEXT CC
190 NEXT CC
200 END
Ready
```

RX-78で出せる8色を行番号130のPAINT文で出してみた。さらに行番号170のCLS文を実行することによって、円の中にぬらされている8色が、どんなふうに変わるかを確かめてみた。色の重ね合わせの逆とはいうものの、意外な色に変わってしまうので、プログラムで活用するときは気をつけたいものだ。

STEP 1 サンプルプログラム-1(まとめ)

```

10 REM ** RX-78 **
20 PRINT CHR$(6) : COLOR, 1
30 LINE[7]32, 16, 14, 16, 14, 72, 21, 72, 21, 48
, 32, 48, 40, 72, 48, 72, 40, 47
40 LINE[7]32, 24, 21, 24, 21, 40, 32, 40
50 CIRCLE[7]32, 32, 8, 1, 3*pi/2, pi/2 : CIRCLE
[7]32, 32, 16, 1, 3*pi/2, pi/2
60 PAINT[7]16, 17, 7 : PAINT[7]42, 71, 7
70 LINE[2]54, 16, 67, 44, 54, 72, 62, 72, 72, 50
, 82, 72, 90, 72, 77, 44, 90, 16, 82, 16, 72, 38, 62,
16, 54, 16
80 PAINT[2]55, 17, 2
90 BOX[4]92, 39, 104, 48
100 PAINT[4]93, 40, 4
110 LINE[6]104, 16, 104, 26, 125, 26, 104, 72,
112, 72, 138, 16, 104, 16
120 PAINT[6]105, 17, 6
130 CIRCLE[5]160, 33, 17, 1, 7*pi/4, 5*pi/4 : CI
RCLE[5]160, 33, 9, 1
140 CIRCLE[5]160, 60, 9, 1 : CIRCLE[5]160, 60
, 19, .9, 2.7*pi/4, 1.3*pi/4
150 PAINT[5]161, 17, 5
160 BOX[2]16, 104, 176, 128
170 CURSOR 3, 14 : PRINT " SUPER PERSONAL
COMPUTER"
180 CURSOR3, 19 : PRINT " BANDAI ELECT
RONICS "

```

注釈文。
画面をすべて消し、バックカラーを青にする。

白で直線8本描く。
白で直線を3本描く。

白でだ円の弧を描く。白でだ円の弧を描く。
座標(16, 17)の区域を白でぬる。座標(42, 17)の区域を白でぬる。

赤で直線を12本描く。
座標(55, 17)の地域を赤でぬる。
緑で直方形を描く。
座標(93, 40)の地域を緑でぬる。

黄で直線を6本描く。
座標(105, 17)の地域を黄でぬる。

水色でだ円の弧を描く。水色でだ円の弧を描く。

水色でだ円を描く。水色でだ円の弧を描く。
座標(161, 17)の地域を水色でぬる。

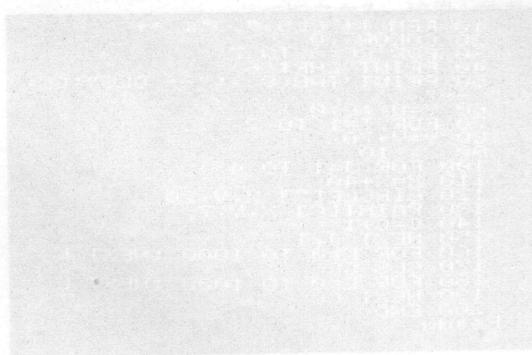
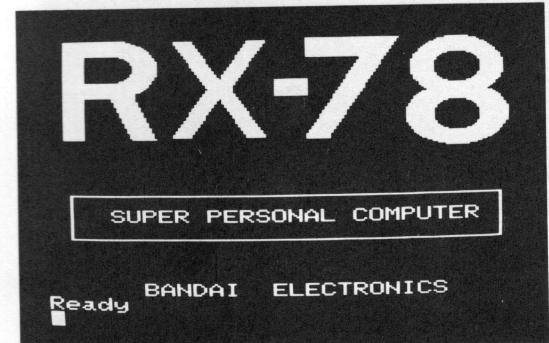
赤で長方形を描く。

カーソルを(3, 14)に移動する。"SUPER PERSONAL COMPUTER"と画面に出す。

カーソルを(3, 19)に移動する"BANDAI ELECTRONICS"を画面に出す。

〔解説〕

LINE文、CIRCLE文、PAINT文、BOX文といったグラフィック関係の命令を用いて画面に大きくRX-78と描いたものです。1つのLINE文でいくつもの直線を描いています。



* 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。

STEP 1 サンプルプログラム-2(まとめ)

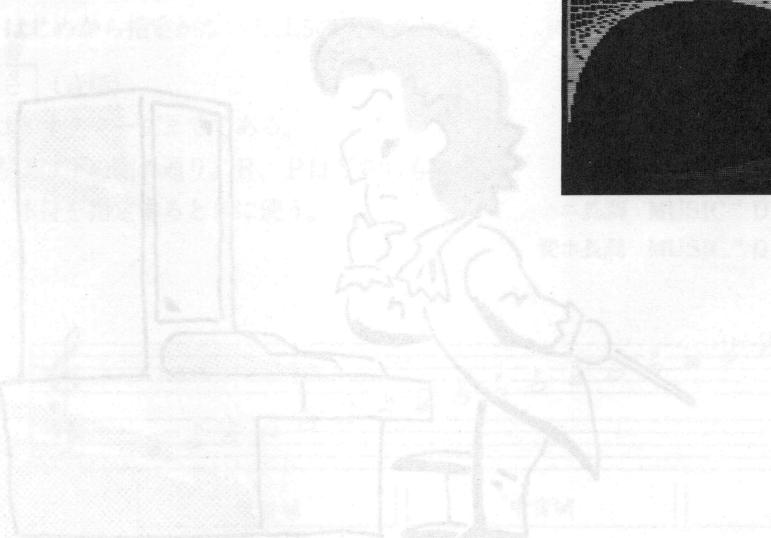
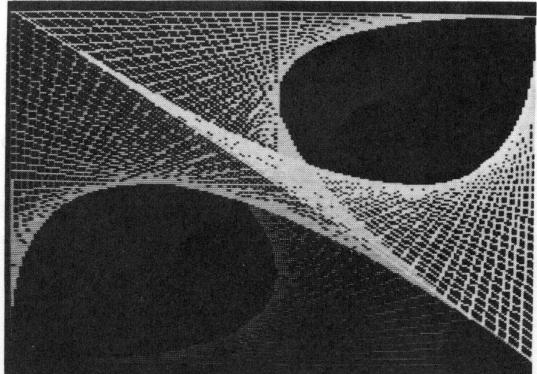
```

10 REM ** ライン フン **
20 PRINT CHR$(6):COLOR5,0 ..... 注釈文。
30 XM=191:YM=183:K=30 ..... 画面にでている内容をすべて消す。
40 XT=XM/K:YT=YM/K:Y=0 ..... XM、YM、Kにそれぞれ値を代入する。
50 FOR X=0 TO XM STEP XT ..... XT、YT、Yにそれぞれ値を代入する。
60 LINE[5]XM-X,0,X,Y ..... Xが0からXMの間、XTだけふえながら、行番号80までを繰り返す。
70 Y=Y+YT ..... 水色でXM-X、0からX、Yまで直線を引く。
70 Y=Y+YT ..... Yの値にYTを加える。
80 NEXT X ..... 行番号50の繰り返し命令の終りを示す。
90 Y=0 ..... Yに値を代入する。
100 FOR X=0 TO XM STEP XT ..... Xが0からXMの間、XTだけふえながら、行番号130までを繰り返す。
110 LINE[3]XM-X,YM,X,Y ..... 紫でXM-X、YMからX、Yまで直線を引く。
120 Y=Y+YT ..... Yの値にYTを加える。
130 NEXT ..... 行番号100の繰り返し命令の終りを示す。
140 X=0 ..... Xに値を代入する。
150 FOR Y=0 TO YM STEP YT ..... Yが0からYMの間、YTだけふえながら、行番号180までを繰り返す。
160 LINE[4]0,YM-Y,X,Y ..... 緑で0、YM-YからX、Yまで直線を引く。
170 X=X+XT ..... Xの値にXTを加える。
180 NEXT ..... 行番号150の繰り返し命令の終りを示す。
190 X=0 ..... Xに値を代入する。
200 FOR Y=0 TO YM STEP YT ..... Yが0からYMの間YTだけふえながら、行番号230までを繰り返す。
210 LINE[6]XM,YM-Y,X,Y ..... 黄でXM、YM-YからX、Yまで直線を引く。
220 X=X+XT ..... Xの値にXTを加える。
230 NEXT ..... 行番号200の繰り返し命令の終りを示す。
240 FOR I=0 TO 500: NEXT ..... Iの値が0から500まで変化させる。
250 GOTO 20 ..... 行番号20にとぶ。

```

[解説]

LINE文をFOR～NEXTを用いて連続的に描いただけですがこんなにおもしろい絵が描けます。



* 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。

STEP 2 未来の音が聞こえないか

RX-78には、3つの音を同時に出したり、ホワイトノイズなどの効果音を出す機能が備わっている。これらの命令を駆使するとクラシックからニューミュージックまで思いのままに音が楽しめる。君が作った曲をRX-78に打ち込んで演奏させてみるのも面白いかも知れない。

STEP 2 サンプルプログラム

```

10 REM ** グリンスリーヴス **
20 PRINT CHR$(60) -----
30 CURSOR10,10 :PRINT " グリンスリーヴス "
40 TEMPO 6 -----
50 A$="L333C+C+A-7D+D+D+5E+E+A-6F#+F#+F#
#+3E+E+E+5D+D+G-7BBB5GGG-6AAA3BBB5"
60 B$="C+C+A-7AAA5AAA-6GGG3AAA5BBE-7GGG
5EEE7R5AAA5"
70 C$="C+C+A-6BBB3AAA5G#G#E-6F#+F#+F#3G#G
#G#5AAA-8ACE8"
80 D$="L555G+G+G-8G+G+G6F#+F#+F#+3E+E+E
+5D+D+B7BBB5GGG-6AAA3BBB5"
90 E$="C+C+A-7AAA5AAA-6GGG3AAA5BBE-7GGG
5EEE8R5"
100 F$="C+C+A-6BBB3AAA5G#G#E-6F#+F#+F#3G#
G#G#5AAA-8C+E+A+7"
110 Z1$="AAA5"+A$+B$+A$+C$
120 Z2$="R4"+D$+E$+D$+F$
130 MUSIC@Z1$ -----
140 MUSIC@Z2$ -----
150 FOR I=0 TO 1000 :NEXT I :GOTO 130 -----

```

注釈文

画面に出ている内容をすべて消す。

カーソルを(10,10)に移動する。"グリンスリーヴス"と画面に出す。

曲の速さを6にする。

グリンスリーヴスの曲のデータを変数に代入する。

データに対する曲を演奏する。

I の値を0から1000まで変化させる。行番号130へとぶ。

〔解説〕

このプログラムはグリンスリーヴスを演奏したもので、たいへんきれいな音楽になっています。



* 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。

コンピュータでMUSIC

ミュージック
MUSIC(1)

使用例

① MUSIC " L1CL2DL3E "

音量1でド、音量2でレ、音量3でミの音が出る。

② MUSIC " L3C7D5E3 "

♪(2分音符)でド、♪(4分音符)でレ、♪(8分音符)でミの音が、音量3で出る。

③ MUSIC M\$

文字変数M \$に指定したなかみに従って、音が出る。

一般型式 **MUSIC " 音の大きさ 音の高さ**

音符の長さ

説明

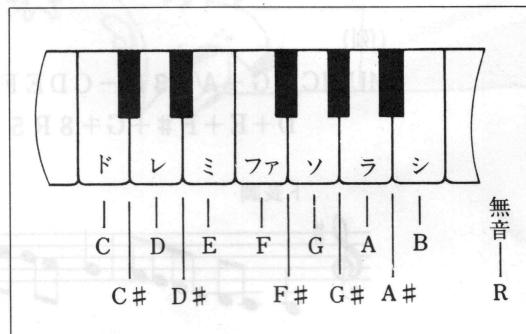
●音の大きさ、音の高さ、音符の長さをストリングあるいは、文字変数で指定すると、RX-78から音が出る。指定の仕方は次の通りである。

音の大きさ (音量)

- 1つの音符ごとに指定できる。
- 大きさの段階は、L0、L1、L2、L3、L4、L5の6段階である。L0は無音で、L2、……と音が大きくなる。
- 指定しないと、その音の前に出た音と同じ大きさになる。はじめから指定がないと、L5の大きさとなる。

音の高さ (音階)

- 高さは3オクターブまでである。
- 指定方法は下の図の通り。R、Pはどちらも無音なので、休符を指定するときに使う。



ハ長調



ニ長調



変ホ長調

(例)
ハ長調 MUSIC " C D E F G A B C + "
ニ長調 MUSIC " D E F # G A B C # + D + "
変ホ長調 MUSIC " D # F G G # A # C + D + D # + "



コンピュータでMUSIC

ミュージック

MUSIC (2)

R0	R1	R2	R3	R4	R5	R6	R7	R8	R9
0	1	2	3	4	5	6	7	8	9

音符の長さ (音長)

- 音符(または休符)の長さは、1音ごとに指定できる。
- 同じ長さの音符(または休符)が続くときは、2音目からは省略できる。はじめから指定しないと、4分音符(または4分休符)となる。
- 指定方法は上の図の通り。

(例)

```
MUSIC " G-A-3 B-C D E F # G 5 A 3 B C +
D+E+F#+G+8 R 5 "
```

ト長調



①

```
LIST REM ** オトナオキサ ***
10 FOR I=0 TO 5
20 PRINT CHR$(62); "L"; I
30 M$=VAL("1STR$(I)+"G"
40 MUSIC M$
50 NEXT I
60 END
Ready
```

行番号50のMUSIC文で、音の大きさをL0～L5までかえてソの音を出している。音の大きさがL0の時は音が出ない。

②

```
LIST REM ** オトナカサ ***
10 FOR I=1 TO 3
20 READ MI$
30 PRINT CHR$(62); I
40 M$=MI$
50 MUSIC M$
60 NEXT I
70 END
80 DATA C-D-E-F-G-A-B-,CDEFGA
B,C+D+E+F+G+A+B+B#+
Ready
```

行番号50のMUSIC文で、音の高さを、1番低い音から1番高い音までの3オクターブを次々と出す。

③

```
LIST REM ** オトナカサ ***
10 FOR I=0 TO 8
20 PRINT CHR$(62); I
30 M$=CHR$(62)+STR$(I)
40 M1$="G"+$STR$(I)+"P6"
50 MUSIC M1$
60 NEXT I
70 PRINT "モード1 やスリリ オカニサテラ"
80 PRINT "CONT ト イレテクラ サイ"
90 STOP
100 FOR K=0 TO 8
110 PRINT K
120 M2$="G0P"+$STR$(K)
130 MUSIC M2$
140 NEXT K
150 MUSIC "G+0"
160 END
Ready
```

行番号50のMUSIC文で、ソの音を32分音符から、付点2分音符まで順に出していく。

行番号90のSTOP文で、プログラムが1度止まる。

CONT [RETURN] といれると、続きをはじめる。

行番号130のMUSIC文で、32分休符から順に付点2分休符の間隔でソの音を出す。

行番号150のMUSIC文で最後に高いソを1音出す。

1人3役 三重和音

ミュージック

MUSIC@

使用例

① MUSIC@ "L333CEG7"

音量3で、2分音符のドミソの和音を出す。

② MUSIC@ M\$

文字変数M\$に指定したなかみに従って、和音を出す。

一般型式

MUSIC@ " 3音それぞれの音の大きさ "

3音それぞれの音の高さ 音符の長さ "

説明

●音の大きさ、音の高さ、音符の長さをストリングあるいは、文字変数で指定すると、最高三重和音までの音が、RX-78から出る。

指定の仕方は次の通りである。

音の大きさ (音量)

●3音別々に指定できる。

●大きさの段階はMUSIC文(105ページ)と同じで、Lに続けて、0~5の数字で指定する。

L [数字] [数字] [数字]
↑ ↑ ↑
1番目の音 2番目の音 3番目の音

指定しないと、その音の前に出た和音と同じ大きさになる。はじめから指定がないと、L5の大きさになる。

音の高さ (音階)

●指定方法は、MUSIC文(105ページ)と同じであるが、必ず3音全部指定しなければならない。

●休符は次の2種類がある。

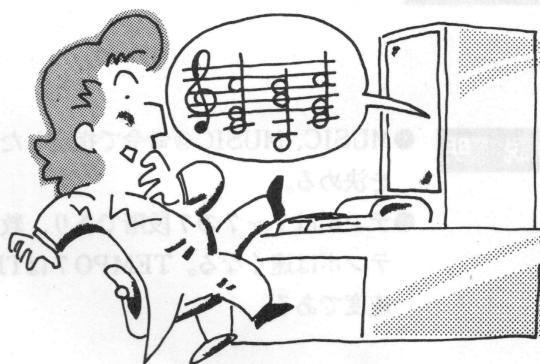
{ R … 3音すべて休み
P … 1音だけが休み

音符の長さ (音長)

●音符(休符)の長さは、和音ごとに指定できるが、和音を構成している各音の長さをかえることはできない。

●指定しないと、その前に出た和音の長さと同じになる。はじめから指定がないと♪(8分音符)の音になる。

●指定方法は、MUSIC文(106ページ)と同じである。



```
LIST REM ** アイサウ **  
10 M1#="L555CEC+8"  
20 M2#="L333G-6B8"  
30 MUSIC@ M1#  
40 MUSIC@ M2#  
50 MUSIC@ M1#  
60 END  
Ready
```

行番号40、50、60のMUSIC文で、おなじみの、「気を付け」、「礼」、「気を付け」の和音を出している。楽符にすると次の通り。



テンポを決める

テンポ

TEMPO

使用例

TEMPO 4

MUSIC、MUSIC@命令で指定した音を、テンポ4（中位の速度）で演奏する。

一般型式 **TEMPO** 演奏のテンポ

説明

- MUSIC、MUSIC@命令で指定した音のテンポ（速度）を決める。
- テンポは1～7の7段階であり、数字が大きくなるほどテンポは速くなる。TEMPO 7はTEMPO 1の7倍の速度である。
- TEMPO文でテンポを指定しないと、TEMPO 4（中位の速度）で、MUSIC、MUSIC@命令を実行する。



```

LIST REM ** TEMPO 1-7 ***
10 FOR I=1 TO 7
20 PRINT "TEMPO="; I
30 TEMPO I
40 MUSIC "CDEFGABP"
50 NEXT I
60 PRINT "モウイチトド:R オワル:E ラオス"
70 INPUT "R テンポ? Eテンポ? ":"A$"
80 IF A$="R" GOTO 20
90 IF A$="E" GOTO 110
100 END
Ready

```

行番号40のTEMPO文で演奏のテンポを1～7まで変化させながら、行番号50のMUSIC文でドレミファソラシの音階を順に出す。

もう1度演奏するか、終わるかを行番号80のINPUT文で聞いてくる。

効果音はシティ感覚

サウンド
SOUND

使用例

- ① **SOUND 0,1** 周期ノイズを、周波数1、音量10で出す。
② **SOUND** 効果音を止める。

一般型式 **SOUND** ノイズモード、周波数
SOUND@ 音量、分周比

説明 ●SOUND文とSOUND@文を組み合わせることによって効果音を出す。

●効果音を止めるには、音の大きさが0になるようにするか、プログラムモード、またはダイレクトモードでSOUNDを実行する。

ノイズモード

●0 周期ノイズ

- 1 ホワイトノイズ
2 TONE 3 出力

のいづれかを指定します。

※ノイズモードで2を指定した場合は周波数は指定しません。

周波数

●0 N/512

1 N/1024

2 N/2048 (N→基本クロック周波数)

3 TONE 3 出力

のいづれかを指定します。

周波数で3を指定した時は、分周比を指定することができます。

0~2は指定できません。

音の大きさ

●0~15の値が指定できます。

●15で音が最大になり、0で音が出ません。

分周比

●ノイズモードで2または周波数で3を指定した時に、指定することができます。

数値または数値変数で指定します。

指定できる値は0~1023の範囲です。

```
10 REM *** コーカオノ サンプル ***
20 PRINT CHR$(6)
30 INPUT" * Push 1:2:3 Key * ";K
40 ON K GOTO 60,110,160
50 GOTO 20
60 REM *** SOUND1 ***
70 FOR S=0 TO 50 STEP .5
80 SOUND 1,3 :SOUND@15,S
90 NEXT S
100 SOUND :GOTO20
110 REM *** SOUND 2 ***
120 FOR S=100 TO 1 STEP -.2
130 SOUND 2 :SOUND@15,S
140 NEXT S
150 SOUND .GOTO 20
160 REM *** SOUND 3 ***
170 FOR S=0 TO 200
180 R=RND(1)*1000+1
190 SOUND2 :SOUND@15,R
200 NEXT S
210 SOUND :GOTO20
220 END
```

行番号30で入力された値によって行番号60、110、160に飛ぶ。

行番号60~100 ホワイトノイズモードによる効果音。

行番号110~150 TONE 3 モードによる効果音。

行番号160~210 RND関数によりランダムな効果音を出しています。

STEP 2 サンプルプログラム

```

10 REM ** オンカラク エンソウ **
20 PRINT CHR$(6).....  

30 COLOR, 1 .....
```

TEMPO 演奏速度

```

40 PRINT" ショパン サッキヨク"
50 PRINT:PRINT" マズルカ"
60 PRINT:PRINT" デス"
70 MUSIC@"DEB3E-G#BG#-EC+CEBA--EA1B-EG#A  
CA3"
80 MUSIC@"B-D#F#A-D#F#A-B-GG-B-F#A-B-E1  
B-B-D#G-B-E3"
90 MUSIC@"E-DBG#-DBFG#BAC+E+1ABD+CEC+CE  
BCEA3"
100 MUSIC@"B-D#F#A-D#F#A-B-GG-B-F#E-B-E  
1F#-CD#G-B-E3"
110 GOTO 70

```

〔解説〕

MUSIC@ を使い 3 重和音でショパンのマズルカを演奏する。三重和音を使う時は 3 音のそれぞれの音量、3 音それぞれの音階を指定すればよい。MUSIC や MUSIC@ で自分で作曲したり、他の曲を演奏してみよう。

注釈文。

画面に出てる内容をすべて消す。

バックカラーを青にする。

"ショパン サッキヨク" を画面に出す。

改行し "マズルカ" を画面に出す。

改行し "デス" を画面に出す。

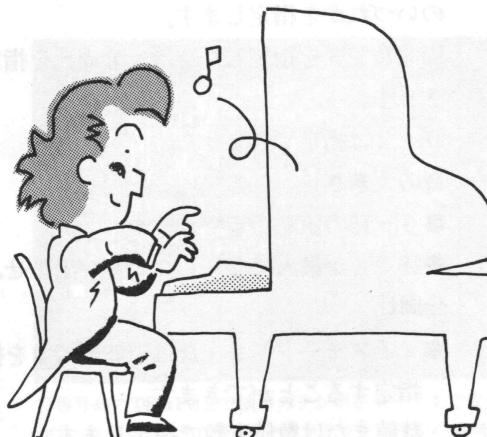
音量 5 で、8 分音符のレ、ミ、シ、8 分音符の低音のミ、ソの♯、シ、8 分音符の低音のソの♯、ミ、高音のド、8 分音符のド、ミ、シ、16 分音符の低音のラ、ミ、ラ、16 分音符の低音のシ、ミ、ソの♯、8 分音符のラ、ド、ラの和音を出す。

音量 5 で 8 分音符の低音のシ、レの♯、ファの♯、8 分音符の低音のラ、レの♯、ファの♯、8 分音符の低音のラ、低音のシ、ソ、8 分音符の低音符の低音のソ、低音のシ、ファの♯、16 分音符の低音のラ、低音のシ、ミ、16 分音符の低音のシ、低音のシ、レの♯、8 分音符の低音のソ、低音のシ、ミの和音を出す。

音量 5 で、8 分音符の低音のミ、レ、シ、8 分音符の低音のソの♯、レ、シ、8 分音符のファ、ソの♯、シ、16 分音符のラ、高音のド、高音のミ、16 分音符のラ、シ、高音のレ、16 分音符のド、ミ、高音のド、16 分音符のド、ミ、シ、8 分音符のド、ミ、ラの和音を出す。

音量 5 で、8 分音符の低音のシ、レの♯、ファの♯、8 分音符で低音のラ、レの♯、ファの♯、8 分音符の低音のラ、低音のシ、ソ、8 分音符で低音のソ、低音のシ、ファの♯、16 分音符の低音のミ、低音のシ、ミ、16 分音符の低音の、ファの♯、ド、レの♯、8 分音符の低音のソ、低音のシ、ミの和音を出す。

行番号 70 にとぶ。



* 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。