

## 漢字の筆順プログラムの作成

—教育学部における初級者のためのマイコン教育用テキストの開発—

大河原 清\*・藤原 昇\*

(1986年1月20日受理)

### Developing a BASIC Program on Stroke-Order in Writing a Chinese Character for Beginners in the Faculty of Education

#### 〔目的〕

本論は、教育学部におけるマイクロコンピュータ（以下マイコンと省略する）利用の教育をするためのテキスト開発の一つの在り方を述べる。開発されたテキストで取り上げる対象は漢字の筆順プログラムである。この漢字筆順プログラムは、教育学部において初めてマイコンに接する学生にマイコンを教える場合に、効率的なマイコン学習のための教材となるものと思われる。ここでいう「効率的」とは具体的には以下の3点を指す。①漢字筆順について学生の抱く考え方を直ちに画面表示できるため、マイコンでのプログラム作成に興味を容易に抱かせることが可能であること。②BASICでの基本的命令語（ここではFOR～NEXT）を使わざるを得ない状況に追い込まれるため、その意味が容易に体得できること。③サブルーチンを導入することで、プログラムの構成を理解できること。

本論の構成は、はじめに教育学部生に初心者マイコン講習会を実施した結果を述べ、次に、その結果から実際の授業場面で実用的なプログラムとして、「漢字の筆順プログラム」をマイコン教育用テキストの題材として選定するに至った状況を述べる。最後に、そのテキスト例の一部と、より進んだ漢字筆順プログラムの1例とを述べる。

#### 〔はじめに〕

教育におけるマイコン利用の教育では、①マイクロコンピュータ用言語、ここではBASIC言語そのものを教える、②マイコンを利用した情報

処理(CMIとしての利用)、③教材を教える(CAIとしての利用)、に大別できる。

このうち、教育学部の学生の教育のためには、上記①②③がそれぞれ重視されなければならない。

\*岩手大学教育学部附属教育工学センター

ところで、第27回国立大学教育工学センター協議会でも西之園晴夫氏が、「確かに、コンピュータ用言語を知らなくても、コンピュータの操作を通して教育におけるコンピュータ利用を教えることが可能ではあったが、やはり有用なプログラムを作成するためには、言語そのものを知る必要がある」と述べておられた。

西之園氏の話からも、学部段階では学生になによりも言語教育の必要性が要請される訳である。しかも、併せてマイコンのCAI的、CMI利用の在り方も追求されなければならないことである。

本論は、こうした学部段階でのマイコン教育用のためのテキストを開発することを目的に、過去のマイコン講習会の経験を下に、こうした目的にかなう教材として漢字筆順プログラムを取り上げ論ずることとする。学習の対象は、マイコンに初めて接する学生である。

#### 〔問題点〕

初心者を対象としてマイコン講習会を実施して強く感ずることは、学習を困難なものとする原因が、ある程度特定できるのではないかと、ということである。それらの原因を大別すると、(1)純粹にマイコン用言語、ここではBASIC言語そのものが難しいために理解できないこと、次に、(2)キーボードでの打鍵方法が困難なために学習ができないこと、最後に、(3)BASIC言語の学習で取り上げられた課題そのものが難しいために学習ができないこと、の三つにあるようである。

こうした考えを持つに至った過去のマイコン講習会について次に述べておこう。

### マイコン講習会実施状況

#### 〔マイコン講習会の目標〕

講習会は、初めてマイコンに接する人を対象に、指導者がいなくても学習者自らが機器に附属する

マニュアルや市販のマイコン入門書を読みながら、BASICについて学習できるようになることを目標としていた。

#### 〔実習内容の選定〕

講習会の目標を達成するため、はじめにキーボードでの英字の打鍵位置を覚えてもらうための、いわゆる英文タイプとしての打鍵の練習をした。次にBASIC言語の学習において自主学习が困難と思われる基本的命令語として、FOR～NEXTといった反復のための制御変数、DIM A(5)等の配列変数を取り上げた。

具体的なテキスト内容の対象は、平均値・標準偏差値の求め方・成績の順位づけのプログラムを作成することであった。

実習内容の詳細は、後述する「表1：実習についてのSD法の結果」の欄の「実習内容」の通りである。実習で使用したテキストはこの通りになっている。つまり、標準偏差値や順位づけを行うために、先に配列変数、反復計算を学習することになっている。これは、はじめてFORTRANを学習する理工科系学生が学習する教材配列に準じたためである。テキストは数式を中心として構成されていると言える。

また、講習会開始に先立ち、キーボードの操作、特にアルファベットの文字位置を覚えていないことが学習を妨げる要因になると予想できたので、講習会ではその練習をはじめに取り入れることとした。

#### 〔使用機器〕

講習会ではNEC製PC-8001(4台)、PC-8001mk11(9台)、PC-9801F(1台)を使用した。

#### 〔実施回数と日時〕

講習会では、機器の数と受講希望者数から、1人1台を使用できるように、昭和59年11月14日

(水)から12月22日(土)まで、水曜日午後3:00～5:00、土曜日午前10:00～12:00の週2回に、受講者34名を対象にこれを2グループに分け、同一内容で実施した。

#### 〔マイコン受講者へのアンケート実施〕

以下は、昭和59年12月22日に実施した『マイコン受講者へのアンケート調査』の結果を中心に述べる。この調査は、昭和59年11月14日から12月22日まで実施したマイコン講習会についての感想を、講習会の最後に調査したものである。講習会へは、29名が持続的に参加したが、完結なアンケート表を回収できたのは次の被験者で述べる通り18名であった。

#### 〔被験者〕

18名。内訳は、教育学部生、2年次1名、3年次6名、4年次9名、研究生1名、現職教諭1名である。性別は、男子3名、女子15名である。

#### 〔アンケート調査結果の見方〕

アンケート調査は、実習内容それぞれについて、好意的評価の最大のもの(1)から困難と評価したものの(5)にわたる5段階のSD法で実施した。表1では、\*印の英語専攻の女学生2名の平均値と、残り16名の学生の平均値を分けて表示してある。特に16名の学生の平均値は、その結果を見易くするため、(1)大変易しい(◎)、(2)やや易しい(○)、(3)普通(□・■)、(4)やや困難(▲)、(5)大変困難(●)といった括弧内の記号で図示してある。特に(3)の「普通」は0.2づつに分割し、3.0～3.4までを(□)で、3.4以上4までを(■)で図示してある。

あくまでも平均値ではあるが、筆者の判定において、◎～●までの数15個が示す通り15段階とし、◎～□の7段階をよい評価、■～●の8段階を悪い評価とする。

更に括弧内の数値は、16名の学生の平均値である。

#### 〔結果と考察〕

結果は表1の「実習についてのSD法の結果」に示す通りである。この表1から分かることは(記号としては、黒い表示)、次の内容の個所でかなり学習が困難となることである。

- (1回目) キーボードでの英字打鍵の練習
- (3回目) 並べ換え(準備)
- (4回目) 反復計算
- (5回目) 配列変数
- (6回目) 得点順の並べ換え

これらは、いずれも講習会実施以前に予想したことを裏付けている。1回目の英字の打鍵練習については、英語を専攻する学生は2名おり残り16名と、理解の面では対照的にその後の学習でも平行線を辿っていることである(表では\*の記号で示してある)。

つまり、英語専攻の2名と残り全員の平均値の変化とを比べてみると、全員の学習を阻害する原因の一つに、明らかに必要な場面で必要な英字を即座に打鍵できないことによる遅れがあることを指適できる。

しかし、4回目からはこの傾向は顕著ではなくなる。この理由は、英字打鍵の習得の良否とは異なる。それは、BASICの命令語の意味理解そのものに関わるものであることが予想される。

この場合、特に考慮しなければならないのは、使用したテキストの良否である。本講習会で使用したテキストについての感想の結果は、表2にまとめてある。

表2における1の「テキスト内容の分かりやすさ」は平均値が1.75と好意的に評価されていた。内容として取り上げた標準偏差値も1.94と好意的評価であった。テキストの記述や実習目標の設定水準も好意的評価であったが、実習内容の程度となると3.50と悪くなるのは、BASIC言語そのものの理解と直接かかわってくるものが予想される。

ここでは、BASIC言語と同時に、プログラムにおける等号の意味の理解も関わっている。つ

表1 実習についてのSD法の結果

回	日時	実習内容	1 大 変 易 し い	2 や や 易 し い	3 普 通	4 や や 困 難	5 大 変 困 難
			◎◎○○○○□□■▲▲▲●●				
1	11月 14・17日	英文ワープロとしての用法 ①英文タイプの打鍵練習・A\$ ②PRINT・LPRINT~ ・END・RUN ③SAVE・LOAD ④NEW・LIST ⑤RENUM	*		■		(3.55)
			*		■		(3.79)
			*	○			(2.87)
			*		□		(3.27)
			*	○			(2.73)
			*	□			(3.00)
2	11月 21・24日	基本的な命令語の学習 ①直接モードによる用法 ②間接モードによる用法 ③LET ④画面の制御WIDTH ⑤ " CONSOLE ⑥ " LOCATE ⑦ " COLOR	*	○			(2.92)
			*	□			(3.07)
			*	□			(3.07)
			*	□			(3.07)
			*	□			(3.07)
			*	□			(3.29)
			*	□			(3.29)
			*	□			(3.00)
3	11月28日 12月 1日	並べ換え(順位と最大・最小値を求め るための準備) ①INPUT ②READ・DATA ③原始力による平均・標準偏差値を 求めるプログラムの作成 ④IF~THEN ⑤GOTO	*			■	(3.67)
			*	□			(3.33)
			*		■		(3.53)
			*		■		(3.73)
			*		■		(3.60)
			*		■		(3.47)
4	12月 5日	反復計算 ①FOR~NEXT ②和を求めるプログラムの作成 (S=0, S=S+N)				■ (*)	(3.50)
						■ (*)	(3.56)
						* ■	(3.81)
5	12月12日	配列変数 ①DIM A (5) 等 ②平均・標準偏差値を求めるプログラ ムの作成 (DIMとFOR~NEXT の利用)				■ *	(3.87)
						■ *	(3.67)
						▲ (*)	(4.07)
6	12月19日	得点順の並べ換え ①FOR~NEXTの二重ループの 用法				■ *	(3.81)
						■ *	(3.81)

- 〔注意1〕 ( ) 内の数値は英語専攻の学生を除く16名の平均値である。  
 〔注意2〕 \*印は英語専攻の学生2名の平均値である。  
 〔注意3〕 ( ) 内の\*印、つまり(\*)は英語専攻の学生の平均値と16名の学生の平均値とが重なることを示す。  
 〔注意4〕 実習内容の欄の○つき数字以外のタイトルについての平均値、つまり「英文ワープロとしての用法」(3.55)や「基本的な命令語の学習」(2.92)などはその回の全体的回答であり、その回毎の○数字の値を平均したものではない。

表2 テキスト内容と実習目標の設定等についての感想

	1 大 変 良 い	2 や や 良 い	3 普 通	4 や や 悪 い	5 大 変 悪 い
	◎◎○○○□□■	○○○○○□□■	○○○○○□□■	○○○○○□□■	○○○○○□□■
1. テキスト内容の分かりやすさ	◎		*		(1.75)
2. 取り上げられている内容そのもの (標準偏差値)	◎		*		(1.94)
3. テキストの記述	◎*				(1.75)
4. 実習目標の設定水準	◎		*		(1.88)
5. 実習内容の程度		*	■		(3.50)
6. 全体としての理解度		○	(*)		(2.53)
7. 機器の操作		○		*	(2.76)

〔注意〕 ( ) 内の数値・\*印・(\*)印の使い方は表1の〔注意1〕～〔注意4〕を参照のこと。

まり、二つの数を交換するプログラムにおける等号が代入の意味であること、1 F A=B T H E N～での等号が数学で使用されるのと同様の使い方であるといったことの違いである。

さらに、反復制御のFOR～NEXTのループ内で制御変数を数式の一部として使用することからくる命令語の意味理解についての混乱である。すなわち、後述する本論のテキストの資料で説明する通り、ループ内に制御変数を組み込むと、本来単に何回反復すべきかを指定するのに制御変数を使用したにもかかわらず、組み込まれた制御変数の増分に注意を引かれ、本来の意味を見失うことである。

### 漢字の筆順プログラムの開発

前述の考察で述べた困難を克服する教材プログラムの開発が要請される。

以下では「漢字の筆順プログラム」を開発例の

一つとして取り上げる。それは初心者を対象として、このプログラムを開発する目的は次の通りだからである。

- (1) 初心者のために、文字位置に注意を喚起させることで、文字位置を覚えさせるための実習がプログラムに含まれていること、これについては、英文タイプとしての打鍵練習として取り入れることも一つの方法であるが、プログラム作成の過程で自然に学習させることが有効と考えられる。

「漢字筆順プログラム」では、グラフィック文字に注目させ、漢字の線の一部を「○、●、◇、▽、□」などで構成させ、グラフィック文字の種類を記憶させること、併せてテキスト文字とグラフィック文字との対応を記憶させることを目標としている。

- (2) 教育学部生を対象としているため、できるだけ小・中学校の授業場面で活用可能なプログラムであること。つまり、必ずしも理科や数学を

専攻することを目的とする学生ばかりではないので、先の講習会でプログラムの対象とした平均値や標準偏差値といった計算を中心とした内容としないこと、が必要である。

特に、社会科であれば各国の国旗の表示、地図記号の表示、理科であれば試験管内部の化学変化での試薬の色の变化、地球の内部構成の地殻・内核・中間核の色分け表示、月の時期による満ち欠け、レンズによる光の屈折、数学なら係数を変えた場合の曲線の変化の表示などなど……教科内容と関連あるプログラムをマイコン教育用として取り上げることが大切であろう。

「漢字の筆順プログラム」は国語科向けと言

えよう。この利用では、小学校低学年の国語の授業で提示すると、児童に筆順についての興味を喚起できるであろう。

(3) しかも、次の段階で標準偏差値や順位づけを学習するのに役立つ、基本的な命令語のFOR～NEXTなどを自然と導入できること。

「漢字の筆順プログラム」では、FOR～NEXTを導入している。

(4) 更に、次の図1に示す通り、将来、CAIのためのプログラム開発に繋がるプログラムであること。

図1において、「CAIのためのプログラム」と「CMIのためのプログラム」とが点線の長

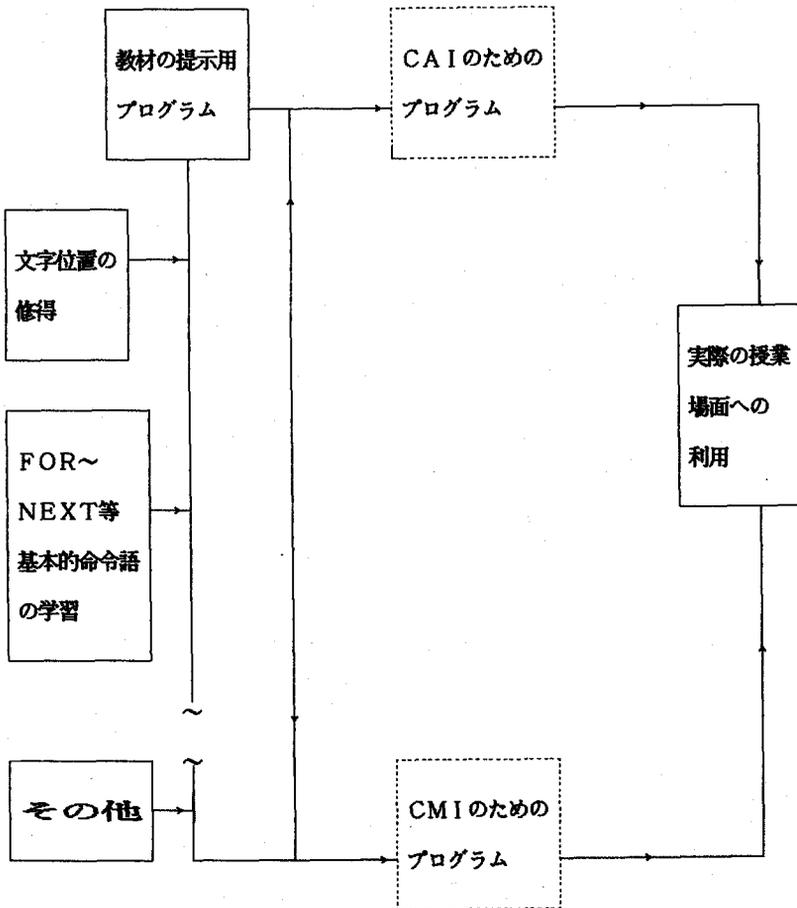


図1 教育学部におけるマイコン教育のためのカリキュラム開発の方向

方形で囲んであるのは、「教材の提示用プログラム」として作成したプログラムを実際の授業場面に直接応用可能ではあるが、それらのプログラムをさらに統合することで、CAIやCMIのためのプログラムとして作成することを学部段階で指導することも可能であることを示している。マイコン学習の中級または上級段階として位置づけられる学習対象と考えられる。

「漢字の筆順プログラム」では、各部首や旁を個々に作成するプログラムを作成しておき、それらの組み合わせから新しい漢字を作るといったプログラムへ結びつけることで、新しい漢字を生成できよう。また、各部首の色分け表示もCAIプログラムとして役立つであろう。

それでは何故このような提示を中心としたプログラム作成を初級者のテキストとして取り上げることが適しているのか。それには幾つかの理由がある。一つはプログラムの作成そのものにかかわることであり、もう一つはその利用にかかわること、更に、教授する立場にかかわることである。

1. プログラムの作成そのものにかかわること  
ここでは、既に実施した標準偏差値・得点順の並べ換えを対象としたプログラムとの比較で考察しよう。

先ず、いろいろなプログラムで、そのプログラムの基本的構成はどのようになっているか、その最も簡単な形がどのようになっているのか。それは、次の「プログラムの基本的構成1」に示す通り、あるデータを入力し、次にそれを出力するというのが最も簡単なプログラムの例である。

〔プログラムの基本的構成1〕

```
10 INPUT "A=" ; A ← 入力する
20 PRINT "A=" ; A ← 出力する
30 END
```

ところが、標準偏差値・順位づけのプログラムは、上記プログラムに計算処理のプログラムが追加された形となっている。つまり、次の「プログラムの基本的構成2」に示す通りである。

〔プログラムの基本的構成2〕

```
10 INPUT "A=" ; A
```

計算処理など、ここに処理のためのプログラムが挿入される

```
20 PRINT "A=" ; A
```

処理の結果を表示するプログラムが、ここに挿入される

```
30 END
```

このため、プログラムで取り上げられる計算処理の方法とその意味についての理解も必要とされる。この増分が、初級者にとっては、BASIC言語の学習の困難度を高めるのである。

初級者のためのマイコン教育のためには、「プログラムの基本構成2」のような計算処理がなく、「プログラムの基本構成1」のように、データなどの入力とその表示・提示を主とすることが重要である。

「漢字の筆順プログラム」は、文字位置の値の入力とその表示のみのプログラムであり、この主旨に適っている。また、グラフィックスとして円や四角形を表示させることも、「プログラムの基本構成1」が主となっていよう。このように、データを入力しそれを表示するようなプログラムの作成、つまり提示用教材のためのプログラム作成の学習が初めには有用であり、計算などの処理を含むプログラムは次の学習段階として考えることが肝要と思われる。一般に、グラフィックスからプログラム作成に入ることが、学習のし易さという点で歓迎されるのはこの理由によろう。

2. 作成したプログラムの利用にかかわること  
第二は、作成したプログラムの利用にかかわる

ことである。

教育学部における初心者を対象としたマイコン教育のためには、マイコン授業で作成するプログラムが実際の授業場面でも直ちに役立つということを自覚させることが重要である。前述した1との関連から、「教材提示用プログラム」の開発がこの目的に適うものである。つまり、実際の授業場面で、マイコンで提示する方が理解に有効な教材を取り上げ、そのプログラムを作成するのである。このため、マイコン教育用テキストの中でも、それらを取り上げることが重要であろう。この方面の利用としては、既に、佐伯卓也氏(文献1)がマイコンの視聴覚的利用として述べられている。この方向は今後、学校現場では広く利用される可能性があるであろう。

### 3. 学習されるBASIC言語が焦点化されていること(教授する立場にかかわること)

第三は、マイコン教育の実習で取り上げるBASIC言語の選定にかかわることである。教授する立場から、1講義時間内の100分程度で、まとまりのあるプログラムの作成が終了することが必要であろう。また、授業で取り上げるBASIC言語の命令語が2~3に特定され、学習の焦点が絞られることが大切であろう。とくに、重要で頻繁に使用される命令語については、各内容のプログラムで重複して使用することが大切であろう。「漢字の筆順プログラム」では、LOCATE, PRINT, FOR~NEXTの三つが中心である。

つまり、上記の指針により開発した漢字筆順プログラムの具体的な特徴は、キーボードの文字操作への注意の喚起、FOR~, NEXTを必然的に利用しなければ作成が容易でないことの自覚、数学的準備の前提を殆ど必要とせず、マイコンを初めて学習し2日目でも容易に取りつき安いこと、実際の授業場面でも有用であること、作成に当たりディスプレイシートでの手作業があること、結果が画面表示され、しかも教育的配慮が生かされ

ることである。たとえば、へんや傍の色別表示が可能であるため、作成者の意図を十分に反映できるなど、プログラムの工夫が大いに可能であることである。

### 〔具体的なテキストの例〕

具体的なテキスト例の一部を後に掲載する。

### 〔実施した結果〕

実際に実施してみると、初めてマイコンに触った学生でも大変興味を持ち、自分が独力でプログラムを作成したという感じを掴ませることができた。特に、予備知識としては、SAVEやLOAD以外、少ない命令語の学習で済むことである。マイコン実習後、3時間目に取り上げても実習が可能であった。テキスト作成後、5名を対象としてマイコン教育を実施した感想は次の通りであった。まず好意的評価はこうであった。(学生A)「一語づつ文字位置を入力すれば、どのように複雑な漢字を誰でも作れると思った。FOR~NEXTやサブルーチンを使うと難しいが楽に字が描けることも分かった。」(学生B)「FOR~NEXTを使わないで字を作るのは手間がかかった。しかしプログラムされた文字が表示された時は満足した。だんだんマイコンの面白さが分かってきた。」次に好意的でない評価はこうであった。(学生C)「自分で座標を設定することがこんなに辛いものだと思わなかった。」(学生D)「今日のプログラムでは、同じことを何度もやっているうちに目だけが疲れてしまった。」

[具体的なテキストの例] .....

## 資料

### 漢字の筆順のプログラムを作成しよう

[BASIC言語学習の中における本項の位置づけ]

この「漢字の筆順プログラムの作成」は、BASIC言語によるマイコン入門において

- ① 基本的命令語（主に、画面制御に関わる）の学習として、
  - ② 漢字筆順に関するCAIプログラムを作成する準備として、
- 位置づけることができます。

[学習目標]

- ① 漢字筆順プログラムの作成を通して、画面の制御の仕方を学習します。
- ② 漢字筆順プログラムの作成を容易にするため、制御変数FOR～NEXTを学習します。
- ③ サブ・ルーチンを学習します。

[具体的学習内容]

具体的に学習する命令語などは次の通りです。

- ① WIDTH
- ② CONSOLE
- ③ LOCATE
- ④ 既に出来上がったプログラムへの新しい命令語の追加
- ④ COLOR
- ⑤ FOR～, NEXT
- ⑥ マルチステートメント
- ⑦ サブルーチン

[指導上の留意事項]

1. 画面の制御としての位置づけ（テキスト画面の座標を知る）

この漢字筆順プログラムの作成は、画面の制御として位置づけることができます。

つまり、グラフィック画面とテキスト画面のうちテキスト画面の学習を中心に進めることができます。グラフィック画面の用法では、点・線・四角形・円の表示をしますが、テキスト画面では、主にグラフィックキーの操作について教授します。

2. 画面の制御に関わる命令語では、次のものを取り上げると有効です。

- ① WIDTH
- ② CONSOLE
- ③ LOCATE
- ④ COLOR

3. 一つづつグラフィック文字を表示する命令を行番号ごとに作成することは、キー操作の不十分な学生にとっては、

文字位置を覚える練習になります。更に、反復制御のFOR～NEXTを導入することで、その使い方を教授できます。

この導入方法は、主に数式での導入、たとえば、和を求めるプログラムでの用法と比較して、容易にFOR～NEXTを教授できます。

座標位置を一つずつ入力するプログラム作成で学生を苦境に追い込み、このFOR～NEXTの命令語が如何に便利であるかを学習させることができます。

- 上記3におけるFOR～NEXTの導入については、FOR～NEXTのループの内部に、制御変数を持ち込まないで、導入することが肝心です。なぜなら、FOR～NEXTの反復制御に着目させることよりも、その増分に学生は注意を向けてしまうからです。→詳細を後述してあるので、参照して下さい。
- 筆順のプログラムの作成が一応完了したなら、この画面制御の発展として、COLORの使い方を導入します。つまり、へんや旁を色分けして表示するため、その画の区切りなどのプログラムの先頭に導入します。

## 1. 漢字筆順プログラムの作成

さて、これからいよいよ漢字の筆順プログラムを作成します。使用する命令語は、後に参考資料で学習する次の命令を使います。

```
LOCATE X座標, Y座標
```

このLOCATEは、PRINT文の前に置くと、PRINTされる内容を指定の位置に置くことができます。ですから、例えば、グラフィック文字の『⊗』を次の図1の通り表示させるには、プログラム1の通り入力し実行すれば良いのです。

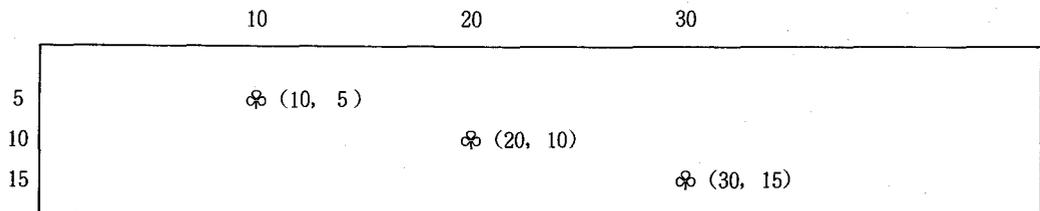


図1 グラフィック文字の『⊗』三つを短時間内ではほぼ同時に表示する

図1のプログラム

```
10 LOCATE 10, 5  
20 PRINT "⊗"
```

```

30 LOCATE 20, 10
40 PRINT "⊗"
50 LOCATE 30, 15
60 PRINT "⊗"
70 END

```

以下でプログラム作成を容易にするため、(1) ディスプレイ・レイアウト・シート、(2) グラフィック文字位置、(3) 同じ形式の命令語の入力方法、について述べます。

(1) ディスプレイ・レイアウト・シート

- ・資料1を参照のこと（ただし本項では省略する）(DISPLAY LAYOUT SHEET)

(2) グラフィック文字位置

- ・資料2を参照のこと（ただし本項では省略する）(キーボードとグラフィック文字の対応図)
- ・GRPHキーを押して使用します。

(3) 同じ形式の命令語の入力方法

〔指導上の留意事項〕

- ・この個所は、時間のない場合には省略して下さい。
- ・図1のプログラムにおけるように、LOCATE 10, 5とPRINT "⊗"の反復した入力、次の通り入力すると便利です。

① 始めは、次の通り入力します。

```

10 LOCATE 10, 5
20 PRINT "⊗"

```

② カーソルを行番号10の十の位に置きます。

```

■0 LOCATE 10, 5
20 PRINT "⊗"

```

③ 十の位の位置で行番号30を作るため、「3」と打鍵します。次の通りになります。

```

3■ LOCATE 10, 5
20 PRINT "⊗"

```

- ④ 上述の状態ではカーソルを右に移動し、座標(20, 10)を指定するため、カーソルを10の十の位に置きます。次の通りになります。

```
30 LOCATE 0, 5
30 PRINT "φ"
```

- ⑤ 上記の状態では、『2』と打鍵します。次の通りになります。

```
30 LOCATE 2, 5
20 PRINT "φ"
```

- ⑥ 同様にして、カーソルを右に更に移動して5の代わりに10を打鍵します。次の通りになります。

```
30 LOCATE 20, 10
20 PRINT "φ"
```

- ⑦ 上記の状態ではRETURNキーを押すと、行番号30の命令が入力されます。次の通りになります。

```
30 LOCATE 20, 10
0 PRINT "φ"
```

- ⑧ 上記の状態では(ピコピコと点滅するカーソルの下には行番号20の『2』が見える)、行番号40を作成するため、ここで『4』と打鍵します。次の通りになります。

```
30 LOCATE 20, 10
4 PRINT "φ"
```

- ⑨ 上記の状態ではRETURNキーを押すと、行番号40が入力されます。次の通りになります。

```
30 LOCATE 20, 10
40 PRINT "φ"
```

- ⑩ 上記の状態ではリストを取って見て下さい。次の通りになっています。

```
10 LOCATE 10, 5
20 PRINT "φ"
30 LOCATE 20, 10
```

```
40 PRINT "⊕"
```

⑩ 上記の状態で行すると、『⊕』が二つ画面に表示されます。

①から⑨の手順を繰り返せば、表示する数が少々増えても苦になることもありません。

ところで、以上のプログラムですと、⊕のマークは実行と同時に画面表示されてしまい、動きがありません。そこで、次の時間待ちの命令を追加します。

#### (4) 時間待ちの命令

```
FOR I=1 TO 500:NEXT I
```

↑  
この数値を変更することで待ち時間を変更出来ます。

図1のプログラムに次の行番号の命令を追加します。

```
35 FOR I=1 TO 500:NEXT I  
55 FOR I=1 TO 500:NEXT I
```

リストを取ると次の通りになります。

```
図1のプログラム（時間を待って表示する）  
10 LOCATE 10 , 5  
20 PRINT "⊕"  
30 LOCATE 20 , 10  
35 FOR I=1 TO 500:NEXT I  
40 PRINT "⊕"  
50 LOCATE 30 , 15  
55 FOR I=1 T 500:NEXT I  
60 PRINT "⊕"  
70 END
```

実行して見て下さい。少し時間がたって二つめと三つめの『⊕』が画面に表示されます。

#### (5) マルチステートメントではコロンの(:)を使う

上述の行番号35や55のNEXTの前にあるコロンの(:)は二つの命令文を一文に書くために使用したものです。

つまり、次の文を一つにしたものなのです。

```
35 FOR I=1 TO 500
38 NEXT I
```

一行にいくつもの命令文を並べたものはマルチステートメントと呼ばれます。

但し一行とは、行の先頭から初めてRETURNキーを押すまでの255文字目迄です。

#### (6) 何回も使用するプログラムはサブルーチンで

上述した通り行番号35や55のように、同じプログラムを反復して使用する場合には、待ち時間をサブルーチンとして使用すると便利です。例えば、次の通りになります。

図1のプログラム(時間を持って表示する)(サブルーチンの使用)

```
10 LOCATE 10 5
20 PRINT "⊗"
30 LOCATE 20 10:GOSUB 1000
40 PRINT "⊗"
50 LOCATE 30 15:GOSUB 1000
60 PRINT "⊗"
70 END
1000 REM *MACHI JIKAN
1010 FOR I=1 TO 500:NEXT I
1020 RETURN
```

行番号30と50のGOSUB 1000の支持で、命令の流れはそれぞれの所で、行番号1000のREM \*MACHI JIKANの所に行きます。次に行番号1020のRETURNによって、待ち時間を実行してから、もとのプログラム本体に戻ってきます。つまり、もともとのプログラムをメインプログラムと呼ぶ場合に、反復して上述のように何度も使用するプログラムはサブルーチン・プログラムと呼ばれます。

#### 例題1

ここで、いよいよ漢字の「一」を図3の通り描くプログラムを作成してみましょう。後述の参考資料2で述べるFOR~, NEXTループも活用しましょう。

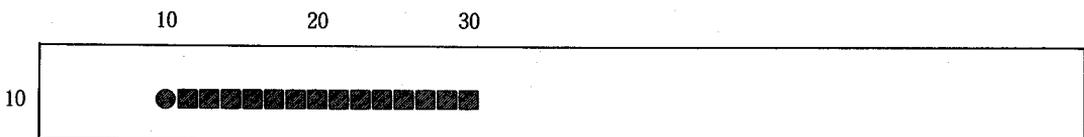


図3 漢字「一」の表示

〔その1：根気よく座標位置を入力する場合〕

この場合には、これまで説明してきた通り、根気よく一つ一つLOCATE、PRINT文を座標位置毎に打鍵します。

図3のプログラム

```
10 LOCATE 10 , 10
20 PRINT "●"
30 LOCATE 11 , 10 : GOSUB 1000
40 PRINT "■"
50 LOCATE 12 , 10 : GOSUB 1000
60 PRINT "■"
.
.
.
99 END
1000 REM *MACHI JIKAN
1010 FOR I=1 TO 500 : NEXT I
1020 RETURN
```

〔その2：FOR～、NEXTループを使用する場合〕

先ず、前述のプログラムにおいて、漢字を画面表示することにかかわっているステートメントはどれかをみます。ここでは、それは、行番号30と40のステートメントなので、

① それらを、反復制御するステートメントFOR～、NEXTループで囲みます。次の通りです。

```
FOR X=  TO
30 LOCATE 11 , 10 : GOSUB 1000
40 PRINT "■"
NEXT X
```

FOR～、NEXTループで囲む

② 次に変化させる変数を決めます。

座標は、行番号30のX座標の11、行番号50のX座標12、・・・と最後にX座標を30とする、つまりXの値を11から一つづつ増やして30まで行ったら終わればよいのですから、前記の制御変数Xの値を次の通りにします。

```
FOR X=11 TO 30
30 LOCATE 11 , 10 : GOSUB 1000
40 PRINT "■"
NEXT X
```

- ③ LOCATEの11の代わりに制御変数を置き換えます。

```
FOR X=11 TO 30
30 LOCATE X , 10:GOSUB 1000
40 PRINT "■"
NEXT X
```

- ④ 最後に行番号をFOR~, NEXTにも付けます。以上を纏めると、次の通りになります。

図3のプログラム (FOR~, NEXTを利用)

```
10 LOCATE 10 , 10
20 PRINT "●"
25 FOR X=11 TO 30
30 LOCATE X , 10:GOSUB 1000
40 PRINT "■"
45 NEXT X
99 END

1000 REM *MACHI JIKAN
1010 FOR I=1 TO 500:NEXT I
1020 RETURN
```

課題1. 漢字の『悟〔る〕』の筆順を画面に表示するプログラムを作成しなさい。

〔手順〕

- ① ディスプレイ・レイアウト・シートに『悟』の漢字を書く。  
\*\*\*注意\*\*\*実際の画面とディスプレイ・レイアウト・シートの大きさは異なるので、比率を考えて下さい。
- ② COLOR命令を用いて、部首毎に色分けして表示してみましょう。
- ③ FOR~, NEXTのループを使用して作成してみよう。

〔参考資料1〕

以下では、漢字の筆順プログラムの作成に関連して、画面の制御方法を説明します。

## 1. 画面の制御

それでは、皆さんの使用している画面について、その制御の仕方を説明します。

~~~~~ テキストとして以下中略 ~~~~~

以下は、テキスト画面を中心に説明致します。それではWIDTH, CONSOLE, COLORの命令を説明します。

## 1. WIDTH

~~~~~ 以下省略 ~~~~~

### 〔参考資料2〕

反復計算はマイコンのおはこ

計算機が計算機らしい機能を発揮するのが、ここで学習するFOR~NEXTです。繰り返し計算をする場合に使用します。

## 1. FOR~, NEXTループ

〔例題1〕

1から10までの和を求めるプログラムを作成しなさい。

```
1  REM 1-10 NO WA WO MOTOMERU
2  REM SAVE "WA001"
10  S=0
20  N=0
30  FOR I=1 TO 10
40  N=N+1
50  S=S+N
60  PRINT "N=";N
70  NEXT I
80  PRINT "S=";S
90  END
```

〔解 説〕

上述のプログラムは、FOR~, NEXTループの比較的理解し易いプログラムです。行番号30のFORで始まる文と、行番号70のNEXT文のループがここでの学習の中心です。制御変数1からはじめて1つつ増加させて10になるまで、つまりこの例題では10回繰り返してこのループを抜け出ることを意味しています。

例題1の実行結果は次の通りになります。

```
run
N=1
N=2
N=3
```

```
N = 4
N = 5
N = 6
N = 7
N = 8
N = 9
N = 10
S = 55
Ok
■
```

この一般形は次の通りです。

```
FOR 制御変数=初期値 TO 最終値 STEP 増分
...
NEXT 制御変数
```

〔注 意〕

- ① 但し、増分が1の場合は、STEP以降は省略することができます。
- ② FORとNEXTの次にくる制御変数は同一の記号を使用します。普通、IやJが良く使用されています。
- ③ 後でも再度述べるのですが、FORとNEXTの行の制御変数にのみ注目することが大切です。ループ内の行番号で同じ制御変数を使用されていても、それに感わされてはいけません。あくまでも、上記の枠で囲んだFOR～、NEXTループでは、FOR～、NEXTの行に書かれた制御変数が初期値から最終値になるまで、反復して実行することに意味があるのです。このことを良く噛み締めておいて下さい。

**FOR～、NEXTは反復実行に主眼あり**

ですから、〔例題1〕において、反復回数を3回にするには、STEPが1の場合には、最終値を3にすれば良いのです。

もう一つの方法は、最終値が10の場合にはSTEPを4にしてもよいのです。

- (何故なら、1回目は  $I = 1$ 、  
2回目は  $I = 1 + 4 = 5$ 、  
3回目は  $I = 5 + 4 = 9$ ・・・10以下である。  
4回目は  $I = 9 + 4 = 13$ ・・・最終値10を越える。

よって、3回ループを反復するには、増分を4とすればよいことが分かる。) プログラムは次の通りになります。

〔例題2〕STEPを代えて反復回数を変更する場合

```
1 REM 1-3 NO WA WO MOTOMERU
2 REM SAVE "WA002"
```

```
10 S=0
20 N=0
30 FOR I=1 TO 10 STEP 4
40 N=N+1
50 S=S+N
60 PRINT "N="; N
70 NEXT I
80 PRINT "S="; S
90 END
```

例題 2 の実行結果は次の通りになります。

```
run
N=1
N=2
N=3
S=6
Ok
```

ところで、制御変数を上手に利用すると、前述したようなプログラムを簡便に書き表すことができます。

### 制御変数の上手な利用方法

(FOR~, NEXTのループの中で、制御変数を使用する。)

例題 3. 1 から10までの和を求めるプログラムを作成しなさい。

```
10 S=0
20 FOR I=1 TO 10
30 S=S+I
40 NEXT I
50 PRINT S
99 END
```

例題 3 では、ループ内の行番号30の中で制御変数 I を使用しています。

〔問題 1.〕

FOR~, NEXTを使用しないで, 1から10までの和を求めるプログラムを作成しなさい。

〔解答 1.〕

```
1  REM 1-10 NO WA WO MOTOTERU (WITHOUT FOR-NEXT)
2  REM SAVE "WA003"
10  S=0
20  N=0
30  N=N+1
40  S=S+N
50  IF N<10 THEN GOTO 30
60  PRINT "S="; S
70  END
```

## より進んだ漢字筆順プログラムの一例

ここでは, より進んだ漢字の筆順プログラムの一例として「飛」という漢字のプログラムを作成した。「飛」という漢字は, 筆順が紛らわしいので選ばれた。

プログラムでは, はじめに「飛」という文字を画面に表示する。次にその筆順を示すようにしてある。

具体的には, 次の①~④に示す通り変更と追加をした。

- ① 漢字の座標点をDATA文で用意し, READ文で読み込んで利用した。〔→700行~900行, 520行〕
- ② あらかじめ学習する漢字(例では「飛」)を画面に表示しておき, 別の色でなぞる方法を取るようになるためにCOLOR文を用いた。〔→100行~260行, 360行, 400行〕
- ③ 復習と確認を兼ねて2回筆順を表示のために, 筆順表示部分をサブルーチン化した。〔→500~590行〕
- ④ 座標DATAは他の漢字に変更することを考慮して, DATAの終わりを-1とし表示サブルーチンでチェックするようにした。〔→910行, 530行〕

ところで, 今後さらにこのプログラムを発展させる方法としては以下のようなことが考えられる。

- ① 学習させたい漢字を自由に変更できるようにするため, 漢字の座標データを一々調べなくてもいいようにする。  
(漢字データ変更のプログラム化)
- ② 表示時間, 表示回数を学習者の能力にあわせて変更可能とする。
- ③ 複数の漢字の中から学習したい漢字を選択して学習できるようにする。

他にも, 漢字を毛筆の字体にすることができれば書道の学習にも応用できるなど, さらにプログラムの発展が可能となるだろう。

キーワード: 漢字の筆順, マイコン教育, BASIC言語, プログラム, マイコン教材

(文献1.) 佐伯卓也, 「教育学部におけるコンピュータ教育——岩手大学教育学部数学科の事例」, 竹之内脩(編)『別冊・数学セミナー6 コンピュータと数学』(日本評論社, 1985年, 20頁), pp. 221-228.

```

1 'save"ヒツシ"エン.2"
10 'ヒツシ"エン ノ レンシユウ
20 WIDTH 40,25:CONSOLE 0,25,0,1:COLOR 7:PRINT CHR$(12)
30 PRINT "コレカラ、ヒツシ"エンノ ヘンキョウヲ シマシヨウ !"
40 GOSUB 600
100 PRINT
110 PRINT "
120 PRINT "
130 PRINT "
140 PRINT "
150 PRINT "
160 PRINT "
170 PRINT "
180 PRINT "
190 PRINT "
200 PRINT "
210 PRINT "
220 PRINT "
230 PRINT "
240 PRINT "
250 PRINT "
260 PRINT "
270 PRINT
280 PRINT "
290 GOSUB 600
300 LOCATE 0,22
310 PRINT " イツカク コトニ ヒョウシ"シマスノテ、ノートニ カキマシヨウ !"
320 GOSUB 600
330 COLOR 6:LOCATE 0,22
340 PRINT " テ"ハ スタート テ"ス。
350 GOSUB 600
360 COLOR 2:GOSUB 500
370 COLOR 6:LOCATE 0,22
380 PRINT " モウイチド" ヒョウシ"シマスノテ、タンカメテ クタ"サイ !
390 GOSUB 600
400 COLOR 7:GOSUB 500
410 COLOR 6:LOCATE 0,22
420 PRINT " オワリ テ"ス。
430 GOSUB 600
440 COLOR 7:BEEP
450 END
500 '***** ヒョウシ" *****
510 RESTORE
520 READ X,Y
530 IF X<0 THEN RETURN
540 IF X=0 THEN GOSUB 600:GOTO 520
550 LOCATE X+10,Y+2:PRINT"0"
560 BEEP 1
570 FOR I=0 TO 10:NEXT
580 BEEP 0
590 GOTO 520
600 '***** シ"カン チョウセイ *****
610 FOR I=0 TO 3000:NEXT
620 RETURN
700 '***** DATA *****
710 DATA 2,1,3,1,4,1,5,1,6,1,7,1,8,1,9,1,10,1,11,1,12,1
720 DATA 12,2,12,3,12,4,12,5,12,6,13,7,14,7,15,7,15,6
730 DATA 0,0
740 DATA 15,1,14,2,13,3
750 DATA 0,0
760 DATA 13,3,14,4
770 DATA 0,0
780 DATA 7,3,7,4,7,5,7,6,7,7,7,8,7,9,7,10,7,11,7,12,7,13,7,14,7,15,7,16
790 DATA 0,0
800 DATA 5,3,4,4,3,4,2,4,1,4
810 DATA 0,0
820 DATA 3,5,3,6,3,7,3,8,3,9,3,10,3,11,3,12,3,13,2,14,2,15,1,16
830 DATA 0,0
840 DATA 1,8,2,8,3,8,4,8,5,8,6,8,7,8,8,8,9,8,10,8,11,8
850 DATA 11,9,11,10,11,11,11,12,11,13,11,14,12,15,13,16,14,16,15,16,15,15
860 DATA 0,0
870 DATA 14,9,13,10,12,11
880 DATA 0,0
890 DATA 12,11,13,12,14,13
900 DATA 0,0
910 DATA -1,-1

```



ト イウ カンシ"テ"ス"