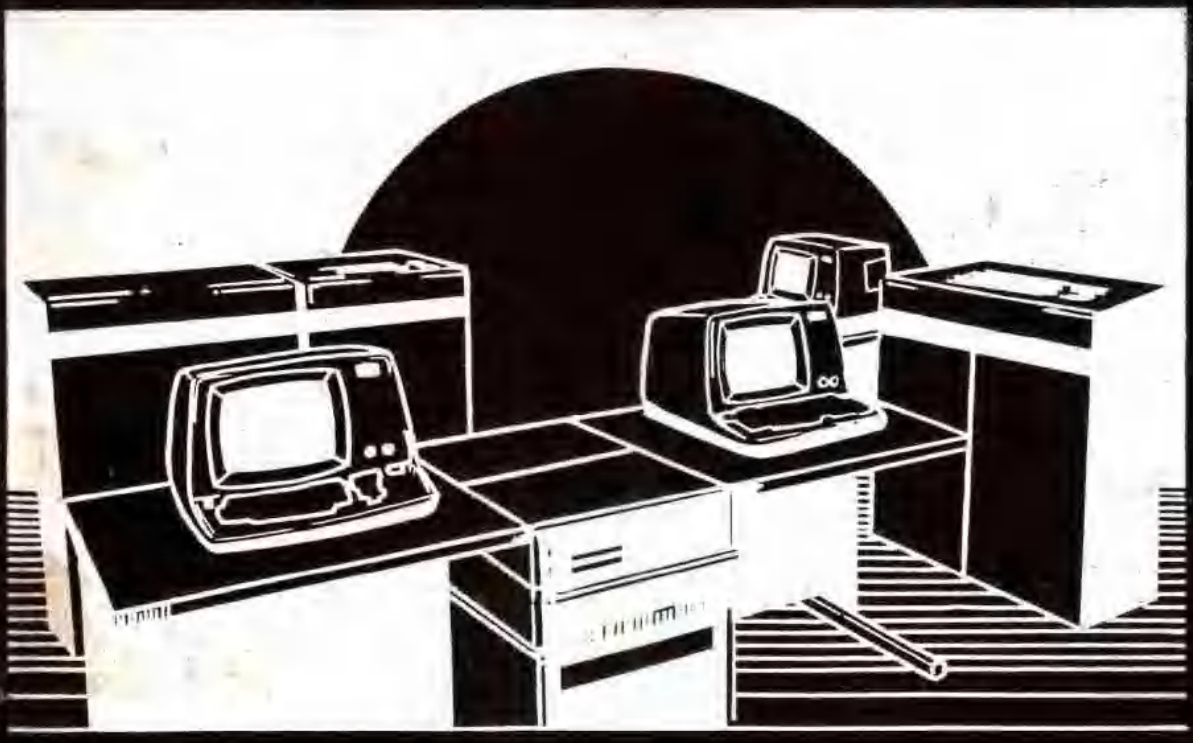
培基語言與應用設計

BASIC PROGRAMMING



周東泉編著

科教出版社

電腦用書

培基語言與應用設計

周東泉 編著

科教圖書出版社印行
總經銷　科教圖書公司

培基語言與應用設計

行政院新聞局局版臺業字第1921號

編著者：周東泉

出版者：科教圖書出版社

社址：臺北市重慶南路1段13號9樓

電話：314-8982

發行者：周杰之

印刷者：漢宮印刷事業有限公司

地址：臺北市金華街223～2號1樓

電話：394-1944

總經銷：科技圖書股份有限公司

地址：臺北市復興南路1段360號7樓

電話：707-3230

特價：三九〇元

中華民國七十一年九月再版

序言

自從電腦發明以來，它應用的範圍愈來愈普及。在三、四年內，電腦將成爲各機關不可少的工具。任何事務處理的最後一步驟會是：如何進行電腦自動化？因此，年青的一代都應學習如何操作電腦以使它協助日常事務上的工作。

如何使電腦動作常會使初學者感到困惑，因爲面對一具精密的電腦是不容易把握住重點。本書的方針乃是教導如何設計程式以指示電腦工作，然而也解釋一些有關必要的電腦結構觀念。近年來各種高階語言有走向完整性的趨向，也就是說：各種語言不限於單獨使用在工程研究或商業中，而可以同時用在工程及企業的領域中。

培基語言是一種最實用易學的語言之一。它不僅可用在工程計算上，並可用於商業上的大量資料處理中。本書的目的就是要使讀者能夠很清楚、正確地使用培基語言，以應用在實際工作上。因此，本書分成兩部份。第一部份是介紹培基語言，並使讀者能夠設計各種程式。例如：第三章將使讀者學會如何設計做資料處理工作的程式。第二部份是在實際應用上的設計研究，分成三章：工程應用、生產管理及商業應用，相信這些資料會使讀者建立起實用上的信心。

本書是在衆多親友的鼓勵及協助下才得完成。在此要特別感謝中心老師賀媛的幫忙整理程式，妹妹蕙娟的協助校稿，及打字人員的辛勞。

若有任何疏漏之處，由作者負全責尚請讀者多予指正是幸。

作者 周東泉 70 9

目錄

1 培基語言概論

培基語言是一種能與電腦交談的多用途語言。在六〇年代中，Dartmouth學院爲了教導各方面學生用電腦處理問題，因而設計此種語言。這種有交談式特性的電腦語言，可使學生在終端機上輸入程式後，與電腦交談。從那時開始，愈來愈多的大學、學院，使用培基語言來教育學生。至今，培基（BASIC）語言在工商業界、大學裡是最多使用的語言之一。

我們可以從圖一看出，培基語言很類似於FORTRAN語言。比較起來，培基語言有一些特點：

1. 交談式—可讓使用的人從終端機上輸入或輸出資料。
2. 檔案處理—可讓使用人建立或更新電腦檔案（包含順序檔及索引檔兩類）。
3. 實用易學的語言—培基語言是一種很容易學習的交談式電腦語言。

由於培基語言在Dartmouth 學院使用成功，如今各廠牌電腦都可供此種語言的使用，例如：IBM、WANG、H.P.，Honeywell，……等。各廠家所提供的培基語言大致相同，但都有各家獨特的性能，例如，WANG的培基語言在交談式的功能上發揮盡致，同時亦可作批件作業（Batch）。我們在這章要介紹培基語言的主要特性，然後再與FORTRAN語言作比較。

1-1 培基語言結構的要素

用培基語言寫好的程式（Program）是由終端機輸入的。輸入的程式可以顯示在終端機的螢幕上或印在報表紙上。程式輸入後，可保留在電腦內；當下一次使用時，可以把它叫出來。

程式是由一些指令（statements）組成的，是我們叫電腦動作（執行、處理）的命令步驟。每個指令開頭有一序號（line number）。由終端機打入指令時，系統就會自動定一個序號（從100，200，300，400，……接續下去）

圖一、一個換算溫度的培基程式範例：

此程式計算從華氏－50度到100度，所對應的攝氏（Celsius）及卡氏（Kelvin）溫度。要做此種換算，應利用底下公式：

$$CEL=\frac{5}{9}(FAH-32)$$

$$KEL=CEL+273$$

此處：FAH表示華氏溫度

CEL表示攝氏溫度

KEL表示卡氏溫度

所寫的培基程式爲：

```
000100 PRINT "FAHRENHEIT","CELSIUS","KELVIN"
000200 FOR FAH = -50 TO 100 STEP 10
000300 CEL = 5 * (FAH-32) / 9
000400 KEL = CEL + 273
000500 PRINT FAH,CEL,KEL
000600 NEXT FAH
000700 STOP
```

以下是執行程式後所印出的報表：

```
FAHRENHEIT        CELSIUS              KELVIN
-50               -45.5555555555555     227.444444444444
-40               -40                   233
-30               -34.4444444444444     238.555555555555
-20               -28.8888888888888     244.111111111111
-10               -23.3333333333333     249.666666666666
 0                -17.7777777777777     255.222222222222
 10               -12.2222222222222     260.777777777777
 20               -6.66666666666666     266.333333333333
 30               -1.11111111111111     271.888888888888
 40                4.44444444444444     277.444444444444
 50                10                   283
 60                15.5555555555555     288.555555555555
 70                21.1111111111111     294.111111111111
 80                26.6666666666666     299.666666666666
 90                32.2222222222222     305.222222222222
 100               37.7777777777777     310.777777777777
```

。這個序號表示電腦執行指令的次序。每指令前、後、中間的空白的個數沒有限制，空白僅表示段落。

指令用到的字母包括A到Z（大寫或小寫），ϕ－9以

及下列特殊符號：

符號	名稱	
	空白	（各特殊符號之用途，
$	金額符號	以後各章介紹）
(	左括號	
)	右括號	
=	等號	
+	加號或正號	
—	減號或破折號	
/	除號	
*	乘號或星號	
**	指數次方	
++	數號	
>	大於符號	
<	小於符號	
↑	竹號	
"	引號	

通常指令是用大寫字母寫成的，小寫字母有時用做輸出的資料。

至於如何在終端機上操作，是利用一些號令（Command）請參閱拙著“操作手册”。

1—2 培基指令的類別

培基指令可分成下列五類，將在本節做一簡介：

1. 資料類 （Data）
2. 運算類 （Computational）
3. 控制類 （Control）
4. 輸入輸出類 （Input／Output）
5. 檔案類 （File）

1-2-1 資料類

資料是我們叫電腦處理（執行）的數據，或處理時所依憑的數據。資料分爲數字資料及文數字資料（alphanumeric data），文數字資料即由任何字母構成的資料。資料可在程式執行時（即電腦開始執行我們寫好的程式）由終端機輸入或由電腦系統中某檔案輸入，或者資料已寫定在程式中。故給予電腦執行時所需的資料有上述三種方式。（檔案可看做儲存在電腦系統中，有固定格式的資料）在程式中，表示資料的方式有二種，即用常數或變數（variable）。以下是培基語言中常數或變數的例子：

常　數	變　數
9502，9502%，2	DER，Y，G1，C5，WANGFL
1.43，−52.1E+2	D%，H9%，U3%
"WR I"	N3$，A1$，ABBA$
	G(2,3)，J2$(1,9)
	ARRAYB(2,3)，ARRAY$(1,2)

從上面的例子，可看出寫常數或變數的一些規則，例如：常數尾部加上"%"的表示整數常數，沒有"%"表示浮點常數；變數尾部有"%"表示整變數（即此變數只能放整數），變數尾部有"$"表示文數變數（即此變數可放文數字，即此變數值可由文數字組成）。變數的名字除第一位外，可由0～9，A～Z組成，第一位只可爲A～Z之字母。

變數是在程式中，用以表示某值。若表示的值是數值，則稱此變數爲數字變數（數字變數分整變數及浮點變數）；若表示的值是文數字值，則稱此類變數爲文數變數。對於文數變數，使用前必須先加以定義其長度，如下例：

```
200  DIM BA$25 , CI$2 ,  K$(3,3)10 , P$(15)6
```

此DIM 指令是專門用在定義文數變數的長度，其中的A$25 定義A$ 變數長度爲25 bytes（即其值至多爲 25 位文

數字），C1$2定義C1$ 變數長度爲2，k$(3,3)10 定義K$()二度矩陣（array，請看1—2—3節 ）的每一元素長度爲10，P$(15)6定義P$()一度矩陣的每一元素長度爲6。

1—2—2

運算類 培基語言的運算指令與FORTRAN差不多，例如：

在FORTRAN中：

Y=(X−A)*W

但在BASIC中：

200 LET Y=(X−A)*W

其實，BASIC 中LET可以不寫，在FORTRAN中序號可以不寫，故兩者極爲相似。上指令的意義爲，把等號右邊的算術式（arithmetic expression)計算後，定爲變數Y的值。所謂算術式是指數個變數或常數以括號或運算符號連結而成。運算符號有下列幾種：

FORTRAN 符號	BASIC 符號	意義
+	+	加
−	−	減
*	*	乘
/	/	除
**	**	次方

底下是一些正確的BASIC指令：

```
100  LET W=X**2+Y
200  LET B9=(A3-A2)**3+(X*Y)/Z
300  LET N=N+A(K)
400  LET C(2,5)=D(3)+9.5
500  LET G=2.5
```

例 同樣的指令可寫成底下的型態：

```
100  LET  W=X**2+Y
```

```
200    LET B9=(A3-A2)                    (第七十二行)!
300          **3+(X
400          *Y)/Z
500    LET N=N+A(K):LET C(2,5)=D(3)+9.5::G=2.5
```

它的意義就是：1.數與數，或文字之間的空白沒有限制

2.若一指令想用數列完成之，在前列的第72行要加驚嘆號。

3.一列中可以寫數個指令，彼此只要以冒號分隔即可，並且可以有空白指令。

但下列的寫法是錯誤的：

```
100    LE
200    T    W=X**2+Y
300    LET B
400         9=(A3-A2)**3+(X*Y)/Z
500    LET C(2,5)=D(3)+9.
600    5
```

其錯誤的原因是指令中的動詞(如LET)、常數、變數應視爲一整體，不可分割成不同位置，故請找出上面程式錯誤之處，一共有4個錯。

下列的程式是一個正確BASIC 列子，請注意其格式上特殊之處：

```
100 X=3:Y=5
200 LET  WASERTT =          X  ** 2 + Y
300   PRINT  WASERTT
310 TRE = WASERTT  * 3
311 B834  = (TRE  -
312      WASERTT) **
313      2
314 PRINT B834
320 PRINT TRE
330 ASSTT1$  =   "123456Y"
340 PRINT ASSTT1$
400 STOP
```

任算術式中運算的次序爲：指數運算、乘除運算、加減運算，由左到右。試觀察下列：

000200 WANG9＝A＊B＊＊5－C＋5.0／X＊Y

及其執行算式 WANG9＝(A)(B^5)－C＋

＋(5.0／X)(Y)

1-2-3

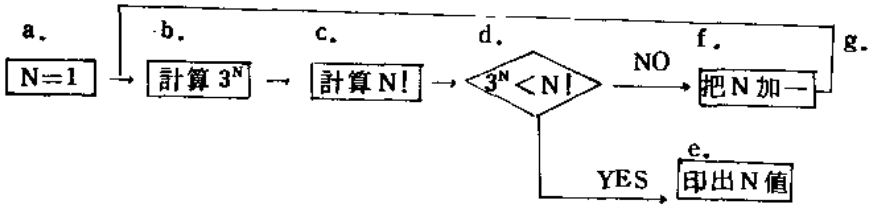
控制類

前面提到：電腦執行你所寫的程式，是按照指令的序號，即小號的先執行，再執行大號的指令。例如前一個例子，電腦先替你執行序號爲000100 的指令，然後再執行序號爲000200 的指令，然後再 000300 號指令，再 000310，000311，000312…；……最後執行 000400 結束。通常是按照這種順序執行的。但某些指令會改變這種順序；換句話說，當它被執行後，會叫電腦去執行某特別位置的指令。我們稱這類指令叫做“控制類”指令。

用在處理邏輯判斷的問題上，這類指令是十分重要的。我們用底下例題來說明控制類指令的特性：

例 求出最小的正整數 N，使得 $3^N < N!$

SOL：爲了要解決這個問題，我們使用“流程圖”來安排求答案的步驟，也就是指示電腦執行的步驟。



每個方塊代表電腦步驟執行的動作，由左到右，顯示出執行的步驟，但是在末兩步驟，需要有選擇的路徑及返回的路徑，這就是控制指令的功能，以下說明所需的控制指令。

在 b、c 步驟中，需用到 FOR及 NEXT兩控制指令，此相當於FORTRAN 語言的 DO 及 CONTINUE 兩指令。其格式如下：

```
000200    FOR   I=1  TO  N
000300    EXPEOFE3 = EXPEOFE3*3      } 步驟 b.
000400    NEXT  I
000500    FOR  J=1   TO  N
000600    LEVELEOFEN=LEVELEOFEN* J   } 步驟 c
000700    NEXT  J
```

步驟 b 中，共需三個指令（200 ～ 400 號），此三個指令反覆執行N次；同理步驟 c 中，亦需三個指令（500 ～ 700 號），要反覆執行N次，而得到N階層，這種返覆的作用即是FOR ～ NEXT 指令的功能。

在 d 步驟中，要比較 3^N 是否小於N!若小於，則執行 e 步驟，否則執行 f 步驟。IF ～ THEN ～ ELSE 指令即提供這種邏輯分支的功能：

```
000800  IF EXPEOFE3 <LEVELEOFEN  THEN 1100
```

此指令中有 1100，即當比較條件成立時，分支到序號 1100 處執行。

步驟 f 、g 即把 N 加一，再分支到步驟 b。這種無條件分支要用 GOTO 指令：

```
000900    N=N+1
000100    GOTO  200
```

步驟 e 是要印出答案並停止執行，這種停止執行要用 STOP 指令：

```
001100    PRINT   N
001200    STOP
```

以上我們從一簡單的例子中，可以瞭解到控制類指令的重要性。以下列舉一些例子說明控制指令的各種格式。

例

```
000100    FOR  K=1   TO  50  STEP  .2
            ⋮
001200    NEXT  K
```

從指令100 到指令1200 反覆做。第一次做時K＝1，第二次做時K＝1.2，第三次做時K＝1.4，……最後一次 K＝50。

例

```
000100   FOR   COUNTER＝3  TO－3   STEP－1
                    ⋮
001100   NEXT  COUNTER
```

此例，每次循環COUNTER 值減一。

例

```
000100  DIM A(3,10)
000200   FOR J＝1  TO 3
000300   FOR I＝5  TO 6
000400   LET A(I,J)＝100
000500   NEXT I
000600   NEXT J
```

此例，有雙重的反覆。內層是從指令300 到指令500 返覆2 次，第一次I＝5，第二次I＝6；內層每反覆二次，外層要反覆一次，因此指令400定變數值的次序爲：A(5,1)，A(6,1)，A(5,2)，A(6,2)，A(5,3)，A(6,3)。

例

```
000100  ON X GOTO ,, 1000 , 2000 ,, 4000 ,,, 3000
```

依X 值無條件分支到不同序號，試看下表

當X爲－2	不分支
－1	不分支
0	不分支
1	不分支
2	不分支
3	分支到 序號1000 指令
4	分支到 序號 2000 指令
5	不分支
6	分支到 序號 4000 指令
7	不分支
8	不分支
9	分支到 序號 3000 指令
10	不分支
11	不分支

例

```
000300    IF  A > 45   THEN  100
000400    IF  WANG$ = " Y "   THEN  1000
```

IF ～ THEN 指令，是個條件分支指令，即條件成立時則分支。300 號指令是一種算術型條件，條件內有算術比較符號——

比較符號	意　義
=	等於
<	小於
>	大於
<=	不大於
>=	不小於
<>	不等於

最後，列一個表說明常用到的控制指令，詳細的應用方法，將於後章敘述：

CALL	FOR …NEXT	INPUT
RETURN	IF …THEN…ELSE	ACCEPT
END	ON …GOTO…	STOP
GOSUB	ON …GOSUB	
GOSUB	GOTO	
FN		

1-2-4 輸入輸出類

BASIC 語言提供了多種輸入輸出資料的指令，本節所介紹的是終端機及印表機（ printer ）的輸入輸出指令；另外有關檔案的輸入輸出指令在下節介紹。

何謂輸入？乃站在電腦的立場來說的，電腦照你的程式執行時，需要有資料輸入或數據輸人，才得執行。輸入的方式有三種：即由終端機或電腦已存之檔案或先寫定在程式內。

何謂輸出？也是站在電腦執行工作的立場來說的，當它在執行你的程式時或執行後，要輸出結果，以讓我們知道它工作後的結論。輸出的方式有三種： 顯示在終端機上或印表機上或檔案中。

首先，我們介紹在程式內如何寫定一些資料？通常少量的資料可用此法，請看

```
000100   READ DATE, RATE, DISCRATE
000300   DATA 1231, 0.012, 0.001
```

READ之意
即讀取資料給變數DATE, RATE, DISCRATE ；
所以把DATA指令所定的值賦予變數。因此DATE的值爲1231，RATE值爲0.012，DISCRATE 值爲0.001。

```
000100   READ  B7, A3
000200   DATA  1.807
000800   DATA  -3.146
```

本例中，變數B7讀取1.807，變數A3讀取−3.146
其次，我們介紹一種最簡單由終端機輸入資料的指令。請看下例：

```
000100   INPUT NAME$, STUDENTID
000200   PRINT STUDENTID, NAME$
000300   STOP
```

本例第100號指令INPUT ，即要求由鍵盤輸入資料給變數NAME$ 及變數STUDENTID 。所以在執行時終端上會出現“？”符號，表示索取資料。順便一提第200號指令PRINT 是把資料印在終端機上，所以是與第100號指令的動作相反，即印出變數STUDENTID及NAME$ 的值。下圖是電腦執行時，終端機出現的半個畫面。