

高級中學科學課程試用教材

電子計算機

第六章 BASIC語言介紹

6-1 — 6-4

6-1 程式語言簡介

人與人之間，不管是直接交談、兩地通訊或知識傳授，都必須透過某種語言溝通。現代人多了一樣有力的工具——電子計算機，也必須透過語言與之溝通，人與機器之間的語言就是程式語言。隨著電子計算機的發展，已經有各式各樣的程式語言在使用。

計算機能夠直接了解的語言是機器語言 (machine language)，它的語句都是由二進位的“0”與“1”所組成的，程式非常瑣碎冗長，平常人不易書寫和閱讀。為了方便書寫程式，前一章已經討論過程式語言必須符號化和口語化，一般把符號化和口語化的語言歸類為高階語言 (high-level language)，以別於低階的機器語言。機器既然無法直接了解高階語言，就必須有一連串的翻譯工作，將高階語言程式轉換成機器語言程式，計算機才能接受而加以執行，也就是說在能夠使用某種高階語言之前，必須在計算機裏建立一種語言處理系統。語言處理系統將在下一章進一步討論，本章主要是介紹高階語言。

高階語言除了容易使用外，它和計算機硬體的關係也比較小，同一種高階語言通常只要做小幅度的調整，就可以在不同的計算機上使用，這很像同一種語言有不同地方的口音一般。

隨應用範圍的不同，而有各種不同的高階語言，有的偏重於數字的運算，常用的有 BASIC、FORTRAN 和 PASCAL 等語言；有的偏重於文字的處理，常用的有 COBOL 和 RPG 等語言，而 PL/I 則兼具兩方面的特性。其他還有許多特殊用途的語言，程式設計者常因工作環境的需要，而選擇合適的程式語言。

BASIC 為 Beginner's All-Purpose Symbolic Instruction Code 的縮寫，是 1960 年代美國 Dartmouth 學院 John G. Kemeny 和 Thomas E. Kurtz 兩位教授發展出來的高階語言。BASIC 當初是設計給初學者使用的，但使用的人日多，應用也愈廣泛，大部分的計算機都提供 BASIC 語言。它的優點是簡單易學，且可用來和計算機交談，是一種交談式語言 (interactive language)。

學習一種語言，最主要的是要了解它的語法 (syntax) 和語意 (semantic)。語法指的是語言使用的規則，也就是文法，程式的文法若有錯誤，就無法翻譯成機器語言，計算機是不會接受的。

語意指的是語言所表達的涵意，有時語法上沒有錯誤，但語意和程式設計者所要表達的有出入，計算機雖然會執行，但獲得的並不是預期的結果，所以語法語意必須兩者兼顧，才能寫好程式。

6-2 程式的操作方法

進入CP/M作業程式，螢光幕上出現交談標誌“A>”後，鍵入

MBASIC ↵

螢光幕會出現一些字樣，說明已經連接好BASIC語言處理系統，以及可使用的記憶容量。當出現“OK”後，即進入BASIC的命令階段（command level），這個階段以“OK”為交談標誌，可用一些BASIC的命令來操作BASIC程式，例如輸入、儲存、更改、顯現、印出、執行等動作，底下將用一個例子來一一加以說明。

BASIC程式每一行前頭都有一行號，介於0～65529之間，由小至大排列。每一行不可超過255個符號，每行可由數個敘述構成，但須以冒號“:”分隔。BASIC程式輸入的方式有兩種，第一種是直接由鍵盤按入，另一種由檔案庫輸入，但若是新的程式，則必須採取第一種方式。讀者可試著將底下程式直接由鍵盤按入：

```
10 INPUT L,M,N
20 IF L<M THEN GOTO 60
30 IF L>N THEN PRINT L: GOTO 80
40 PRINT N
50 GOTO 80
60 IF M>N THEN PRINT M: GOTO 80
70 PRINT N
80 END
```

在輸入程式之前，若先鍵入命令“AUTO ↵”，則以後行號不用打會自動產生。程式輸入完畢後，按 **CONTROL** 鍵螢光幕會出現交談標誌“OK”。若再鍵入

RUN ↵

程式開始執行，然後出現問號“？”，表示需要資料，因為在行號為10的敘述裏需要輸入資料L，M，N，若打入三個數目，程式繼續執行，最後會將輸出的資料顯現在螢光幕上。

當認為程式沒有錯誤，想保存起來以便以後再使用時，可鍵入

SAVE “檔案名稱”，A ↵

整個程式就以該檔案名稱存於檔案庫中，注意這裏的檔案名稱要用雙括號“ ”括起來，A表示程式是以ASCII碼存檔。

若要檔案庫輸入這個程式檔案，可鍵入

LOAD “檔案名稱” ↵

如此則該程式檔案已被找出。若鍵入的是

LOAD “檔案名稱” R ↵

程式被找出後，就直接執行，而不必再鍵入執行命令RUN。

當程式已經由鍵盤或檔案庫輸入，要查看時可用命令

LIST ↵

整個程式便再度顯現在螢光幕上。LIST後面若加上行號，則只顯示出那些行，例如按LIST 40則只顯示

```
40 PRINT N
```

按LIST 30-50 則顯示出

```
30 IF L>N THEN PRINT L: GOTO 80
40 PRINT N
50 GOTO 80
```

程式在偵錯時常要加以修改，有三種簡單的方法可以運用：重寫、插入或刪除。若要重寫某行的話，只要重新由鍵盤按入那行即可，例如按

```
40 PRINT "THE MAXIMUM IS",N
```

原來的那一行就被取代，若再按LIST 螢光幕會出現

```
10 INPUT L,M,N
20 IF L<M THEN GOTO 60
30 IF L>N THEN PRINT L: GOTO 80
40 PRINT "THE MAXIMUM IS",N
50 GOTO 80
60 IF M>N THEN PRINT M: GOTO 80
70 PRINT N
80 END
```

若想在原來程式裏插入某些行，也是直接由鍵盤輸入那些行即可，但行號的大小必須符合位置，例如鍵入

```
65 PRINT "THE MAXIMUM IS"
```

這行就插入 60 和 70 兩行之間。鍵入

```
DELETE 行號 ↵
```

會把程式中該行刪除，例如鍵入DELETE 40，則把行號為 40 的敘述去除，如鍵入DELETE 20—70，則把行號為 20—70 的敘述全部去除。

經過練習操作熟練後，坐在終端機前寫程式、執行程式或修改程式，是很方便的事，但若要将程式由列表機印在紙上，則必須鍵入

```
LLIST ↵
```

第一個L代表列表機（Line printer）。

在程式操作過程中，無論程式是由鍵盤直接輸入，或是由檔案庫LOAD 進來，終端機每次只能處理一個程式，若要更換程式，可從檔案庫LOAD 另一個程式進來加以取代，或鍵入

```
NEW ↵
```

再重新鍵入程式。

使用SAVE 命令可建立新的程式檔案，同時也可取代檔案庫中同樣名稱的舊有檔案。從檔案庫中LOAD 一個程式檔案進來，加以修改後，若沒有用SAVE 命令重新存檔，檔案庫所存的仍然是舊檔案。如果要更改檔案庫裏檔案的名稱，可鍵入

```
NAME "新檔案名稱" AS "舊檔案名稱" ↵
```

如果要查檔案庫目錄，可鍵入

FILES ↵

最後要特別記住的是，使用者在BASIC命令階段，必須鍵入

SYSTEM ↵

才能把所使用的檔案關閉，而回到CP/M作業系統。

6-3 常數與變數

BASIC的常數有字串、整數、定點數、浮點數、八進位數和十六進位數等。字串必須以雙括號括出，最長不可超過255個字符。整數必須介於-32768~32767之間。

定點數和浮點數分單精確度和雙精確度兩種，單精確度有效數位為6，而雙精確度有效數位為16。

(A) 單精確度：定點數7位以下或字尾附加“!”號；浮點數以E表示。

例： 1492

24.6!

-8.06E-05

(B) 雙精確度：定點數8位或8位以上，或字尾附加“#”號；浮點數以D表示。

例： 345.62817

64.8#

-1.0432D06

變數的符號為一字串，但必須以大寫英文字母開頭，BASIC語言內所有的命令、敘述、函數、運算裏已經採用的字叫做保留字，保留字因為已經有特定的意義，不可以拿來當變數的符號使用。

程式裏的每一個變數，應看做某項資料的代號，資料的種類必須分辨清楚才能正確的處理，變數如果加上一些特定記號，就可以宣告(declare)變數的資料型態(data type)

附加記號	資料型態
\$	字串
%	整數
! (或不加)	單精確度浮點數
#	雙精確度浮點數
例： PI #	宣告為一雙精確度值
A!	宣告為一單精確度值
NUMBER%	宣告為一整數值
TEMP	宣告為一單精確度值
STR \$	宣告為一字串值

6-4 基本敘述

6-4-1 LET 敘述

程式裏變數的值，有時是直接輸入的，有時是設定或計算的結果，達成後一種效果的敘述，稱為指定敘述 (assignment statement)，形式為

LET 變數 = 式子

此處“=”不是數學上的等號，而是把右邊式子的值指定給左邊變數的意思，其中式子部分可以是常數、變數或是一個運算式子。如果把敘述裏的符號“LET”省略，意思仍然一樣。

例。

<pre>100 LET A%=12 110 LET SAVE=X 120 LET X=X+1 130 LET F=C*9/5+32</pre>	或	<pre>100 A%=12 110 SAVE=X 120 X=X+1 130 F=C*9/5+32</pre>
--	---	--

第一行把整數常數存入變數A%，第二行把變數X的值存入變數SAVE，第三行把當時X的值加1，存入同一個變數X，也就是說X的值已經變成原來值加1。第四行把“=”號右邊按照當時C的值加以計算，計算的結果存入變數F。

在指定敘述裏，要指定給左邊變數的值必須和左邊變數的型態一致，例如

100 LET A%=12

這一行，因A%是整數，12不必轉換資料型態就可指定給A%，而

100 LET A=12

A表示單精確度浮點數，計算機要將12轉換成浮點數，然後才存進A裏。又如

130 LET F#=C*9/5+32

右邊計算的結果是單精確度浮點數，也必須轉換成雙精確度浮點數才存入F#。

指定敘述左邊變數的值會被改變，而右邊出現的其他變數的值都不改變。

6-4-2 IF-THEN 和 GOTO 敘述

指定敘述表達了計算機的基本計算能力，這些能力必須作適當的安排，以構成計算步驟的流程，才能井然有序的處理資料，條件分支和無條件分支就是控制程式流程的敘述。條件分支用IF-THEN敘述，它的形式是

IF 關係式子 THEN 敘述序列 ELSE 敘述序列

它的語意是：如果該關係式子成立，就執行THEN後面的敘述系列，否則就執行ELSE後面的敘述序列。敘述序列可包括一個或一個以上的敘述。這一行執行完畢後，如果沒有跳行的情況發生（即敘述序列裏沒有無條件分支），就接著執行下一行。

例。300 IF X>Y THEN Z=X-Y ELSE Z=Y-X

若X的值比Y大，X-Y的計算結果存入E，否則Y-X的計算結果存入Z。這一行執行後，Z含有|X-Y|的值。

IF-THEN 敘述 ELSE 部分可以不同，形式變成

IF 關係式子 THEN 敘述序列

語意是：如果該關係式子成立，就執行 THEN 後面的敘述，否則這一行算是結束，接著執行下一行。

例。

```
500 IF X<0 THEN X=-X
600 - - - - -
```

若 X 的值為負，先把 X 的值變號，再執行 600；若 X 的值為 0 或正，就直接執行 600。

無條件分支 GOTO 敘述，其形式是

GOTO 行號

表示無條件的跳到指定的那一行。

例。

```
10 IF AMOUNT>3000 THEN GOTO 40
20 LET TAX=300
30 GOTO 50
40 LET TAX=300+(AMOUNT-3000)*RATE
50 - - - - -
```

若 AMOUNT 大於 3000，用 40 計算 TAX，否則執行 20，將 TAX 設定為 300，30 是一個 GOTO 敘述，表示執行 20 後無條件的跳到 50。10 為一 IF-THEN 敘述，其中 THEN 部分用到一個 GOTO 敘述。上面的程式可以改寫成

```
10 IF AMOUNT<=3000 THEN TAX=300: GOTO 50
40 LET TAX=300+(AMOUNT-3000)*RATE
50 - - - - -
```

10 的 THEN 部分用了一個指定敘述一個 GOTO 敘述看來相當簡潔，但一般若計算步驟多的話，前一種寫法段落比較分明。

IF-THEN 敘述和 GOTO 敘述配合使用，就可導出所有可能的流程控制。步驟的重覆執行，在程式裏就是迴路，由底下一個例子來看迴路是如何寫成的。

例。給一個正整數 n ，求平方小於等於 n 的最大整數 M 。

以變數 $N\%$ 代表 n ，變數 $M\%$ 代表 m ，起先 $M\%$ 設定為 1，檢查其平方是否小於等於 $N\%$ ，若是，則將 $M\%$ 加 1，重覆同樣的步驟，直到 $M\%$ 大於 $N\%$ 。首先把這個演算法則寫下來：

$M\%$ 設定為 1

若 $M\%$ 的平方小於等於 $N\%$ 則重覆下面步驟：

$M\%$ 更改為 $M\%+1$

$M\%$ 更改為 $M\%-1$

接著再把演算法則寫成程式：

```
100 LET M%=1
200 IF M%^2>N% THEN GOTO 500
300 LET M%=M%+1
400 GOTO 200
500 LET M%=M%-1
```

400 跳回 200 就形成了一個迴路，300 就是被重覆的步驟。執行 200 時若 $M\% \wedge 2 > N\%$ 就跳到 500，而不再進入迴路。出了迴路後，因為 $M\%$ 的值太大，所以減 1。 (待續)