PC BASIC與 MBASIC

檔案程式手冊

張其邦譯

電腦中心出版

PC BASIC與 MBASIC

檔案程式手冊

張其邦譯

電腦中心出版

PC BASIC 與 MBASIC
檔案程式手册

一般的BASIC 語言課程，大多是將BASIC 各敍述講述完後即算功德圓滿，大功告成。對資料檔案程式方面多略而不提，或草草講述，提早收尾。但對BASIC 語言而言，應用最廣，最有效用的部份卻是資料檔程式的範圍。但是一般的BASIC 語言教科書對這方面所談及的內容不多，專書討論的更少，以至於一般學子要靠有限的文獻，在這方面的領域苦心摸索，其學習的過程自然是漫長而艱苦。尤其是以理工科領域的同學，對檔案幾乎毫無觀念，在學完BASIC 語言後，仍難以一窺資料檔之廟堂，以至於對計算機的應用能力有限，除了一般的科學運算，就再也無法作其它方面的用途了。

本書原係LeRoy Finkel及Jerald R.Brown 兩位所著之"DATA FILE PROGRAMMING IN BASIC"，係針對BASIC 語言之順序檔及隨機檔之討論及教學而編著。全書內容簡潔實用，循序講授，是討論BASIC 檔案程式的最佳教材。另外有鑑於目前BASIC版本衆多，各有所長，而檔案指令也是各BASIC 版本中變化最大的地方，因而以程式的互容性爲主要考慮因素，採取目前使用量最大的MBASIC 爲主，並以IBM個人電腦爲準，重新編寫該書。依目前的情勢推斷，IBM個人電腦較有機會成爲個人電腦的主流，而MBASIC將因而成爲BASIC 的主流語言。所以這本討論BASIC 資料檔程式的教材，其內的BASIC 程式均以IBM PC BASIC及MBASIC 爲準，（IBM PC BASIC是對MBASIC作了某些功能的擴充而成）。並加入了第六章的磁帶順序資料檔一章。

全書的編排均以自修的方式逐項解說，並以誘導的方式進行各習題的練習，所以深信這是一本最具學習效果的一本書。本書可做爲自修手冊，或是高等 BASIC 語言之課本供大專院校或高級學校之程式語言課使用。或是各大學之BASIC 語言實作作業手冊。

如何使用本書

這本書是一本自修手册，一步一步的教導你如何編寫BASIC 資料檔案程式。每章中的每一節都有一個學習主題，以及練習用的問題，和要求你實作的程式習題（自我測驗）。所以使用這本書最好的方式是確實的思考並回答書中的每個問題，並且對每個程式習題編寫程式，而後上機實驗，只有在練習中才能獲得經驗，和體驗書中的說明。編寫資料檔案的程式並不難，只要能掌握幾種基本技巧，所有的問題便能迎刃而解。所以爲了幫助你能確切的掌握資料檔程式的幾個基本技巧，各節的問題都是循序漸近的，而各程式也大多是前後章相聯，慢慢的擴充的。讀這本書不要急躁，不要貪快。按部就班，照著書中的指點一步一步的來，所得的成效最爲確實，最爲有用。

寫程式是件很現實的事情，別人可不管你懂多少理論或看過多少本書，或是這個程式花了多少時間寫成的，唯一評判的標準就是你寫的程式是否能正確的執行某項工作。這就是所謂的刀下見功夫了。所以學習寫程式，實際的程式演練最爲要緊，千萬不要光學光看而不寫。尤其是資料檔案程式，幾乎與資訊的應用上脫不了關係。任何實用一點的程式都必須使用到資料檔案的方式。所以除了會寫一般的BASIC 程式還不算，也不夠用。BASIC 眞正精彩的部份是在BASIC檔案程式中。你可以把BASIC 資料檔程式看作是高等BASIC 來學，所以在學習本書之前，最好已經先完成了普通BASIC 語言之課程，再藉著本書作爲課本或自修教材。

除了基礎的BASIC 語言能力外，就像我們前面所講的，任何程

式工作，實作才是眞正的重心所在，所以我們在各章的自我測驗中，充份的提供了練習的問題，這些練習題的存在，有著多重的意義，透過這些練習你可以對書中的說明有一個實際的驗證機會。透過這些練習，供給你一個自習的過程。也提供你在程式寫作上的實際經驗，這一點是任何教課書都無法供給的。所以我們特別強調，對這本書的例題和習題我們要求每一題都要實作，尤其是透過計算機的實作。諸位讀者最好坐在計算機前來讀這本書，把這本書和計算機融爲一體，而本書的習題設計也正是對這種學習方式而設計的。你將能體會到BASIC資料檔程式眞的一點都不難，只是普通BASIC 的延伸而已，就像一年級讀完了讀二年級一般那麼自然。但是你若眞的作過本書的習題，你所獲得的成就將是難以說明的。你能體會出你目前的能力和從前是多麼的不同，就如同從幼兒脫變成成人一般，眞正的是可以編寫實際有用的程式了。所以別忘了把本書內的各例題和習題拿到計算機上去實作驗證。

從第 4 章的習題開始，將要求你編寫程式以產生一些資料檔，這些資料檔從此以後將貫穿著後面幾章所有的習題，我們透過這些資料檔來學習操作資料檔的各種方法，它是一個基本對像，所以學到那裏的時候別跳過去。耐著性子做到第八章完，到了總結束練習時，你將發現所有的問題都能輕易的解決。

寫程式是很現實的工作，它要靠實力來完成，而實力獲得只有靠練習，尤其是在計算機上的練習。祝你成功。

目 錄

第一章 寫出清晰，易讀又合邏輯的BASIC程式…… 1

簡 介…… 1
BASIC語言 …… 2
應該如何使用BASIC語言 …… 3
寫個易讀的程式…… 4
上而下的組織…… 5
REMARK敍述 …… 5
GOTO敍述…… 7
導引模組格式…… 8
導引模組後的模組…… 10
副程式…… 11
外觀的編排…… 13
間隔（Spacing）…… 13
增加美觀及易讀性的其它技巧…… 15
解除美化規則來節省記憶空間及加快執行速度…… 17
自我測驗…… 19
解 答…… 21

第二章 BASIC敍述重點覆習…… 23

簡 介…… 23
變數名稱…… 23

字串變數…… 25
READ-DATA指派敍述 …… 28
LINE INPUT敍述…… 34
序連（CONCATENATION）…… 35
IF…THEN敍述 …… 36
IF…THEN對字串及ASCII碼之比較 …… 40
LEN函數 …… 44
字串函數：使用字串資料最有用的工具…… 46
利用INSTR作字串的搜尋 …… 51
利用ON…GOTO之多重分支（MULTI-BRANCHING）…… 54
FOR…NEXT敍述 …… 55
多重敍述列…… 57
自我測驗…… 59
解　答…… 61

第三章　建立資料登錄及錯誤檢查常式…… 63

簡　介…… 63
資料欄之長度…… 65
檢查資料長度的登錄…… 68
填補空位——使輸入資料符合正確的欄長度…… 69
消除各欄中子字串的填補空位…… 72
檢查空字串的輸入…… 73
替換資料欄內的資料項目…… 77
VAL函數於資料登錄的使用 …… 79
使用STR $將數值轉換爲字串 …… 81
檢查不合法的字元…… 82
資料登錄與檢查程序之討論…… 84

自我測驗…… 89
解　答…… 91

第四章　建立和讀取順序資料檔…… 93

簡　介…… 93
使用磁片儲存資料…… 94
順序資料檔與隨機接達資料檔…… 98
順序資料檔之初設（INITIALIZING）…… 99
緩衝器的問題……104
印（PRINT）或寫（WRITE）資料進入順序資料檔……105
從檔內讀出資料……114
從磁碟上永久性的消除檔案……118
自我測驗……124
解　答……140

第五章　順序資料檔公用程式…… 149

複製一個資料檔……149
在順序檔尾部增加資料……153
更改檔內的資料……156
檔資料的編輯，刪除與插入……166
合併兩檔的內容（MERGE）……178
順序資料檔之注意事項……189
信件處理程式……192
自我測驗……199
參考解答……203

第六章　卡式磁帶資料檔…… 207

簡　介……207
卡式磁帶資料檔的讀入及寫出動作……209
自我測驗……225
參考解答……232

第七章　隨機接達資料檔……237

簡　介……237
什麼是隨機接達檔？……237
隨機接達檔的初設……239
緩衝器欄位……240
隨機接達檔對字串之讀寫作業……243
隨機接達檔內之數值資料……254
隨機接達檔公用程式……259
轉換順序檔成爲隨機接達檔……267
改變隨機接達檔內各記錄之位置……271
顯示隨機檔之通用程式……272
自我測驗……274
解　答……281

第八章　隨機接達檔之應用……289

順序指標檔……289
個人資金管理應用程式……295
自我測驗……305
參考解答……309

總結練習……313

解　答……319

附錄A　BASIC敍述及函數之使用說明 …… 323

附錄B　ASCII字碼對照表 …… 327

第一章
寫出清晰，易讀又合邏輯的BASIC程式

目標：當您完成本章後，您將能夠作到以下幾點：

1. 瞭解一個程式如何能用上而下（ Top-To-Bottom ）的格式寫成。
2. 利用REMARK敍述（ REM ）寫出導引模組（ Introductory Module ）。
3. 使用六種列印規則，使您的輸出格式清爽悅目。
4. 認識七種能節省記憶體空間的程式方法。

簡　介

這本書教導如何編寫 IBM PC或類似之計算機之資料檔案之處理程式，但您必須已先完成了BASIC語言的基本課程，能看得懂一個BASIC語言程式，及能夠寫一些簡單的BASIC程式。所以這本書不是針對BASIC語言初學者而寫的，而是爲那些已學過BASIC，但還未使用過BASIC資料檔案的人編寫的。資料檔案之程式是以MICROSOFT公司所發展的MICROSOFT BASIC爲主，使用MICROSOFT BASIC 語言的系統或機種計有CP/M系統。（MBASIC或BASIC-80），IBM個人電腦，TRS-80 DOS，及ALTARI 4.1 BASIC等。

在別的BASIC版本內，其資料檔案之敍述（ STATEMENT ）雖不儘相同，但結構都很相似。所以，將本書與您的計算機系統之參考手册合併使用

，將更能發揮這本書的功效。

因爲我們假設您已具有編寫BASIC程式的基本知識，也曾經自己寫過程式，所以下一步爲您加强的是程式的組織和它的清晰度。因爲一個牽涉到檔案的程式可能十分龐大而且複雜，爲了要使程式能正確的運轉，其不可或缺的除錯過程，也就相對的更複雜，更困難。因此這一章對您十分重要。它提供了組織化程式的編寫方法，幫助您在將來能輕鬆愉快的編寫程式。

BASIC語言

BASIC計算機語言是於60年代的早期由Dartmouth學院所發展的。原只是爲了給沒有或只有一點計算機經驗，而且只稍具有一點數學知識的人所設計的。它原來的語法只包括了初學者所需要的功能。但當一些學院、計算機廠商及一些機構接受了BASIC語言後，又各自在其中填加了一些功能來配合他們的需要。所以BASIC語言出現了許多不同的版本，如Extended BASIC，Expanded BASIC，SUPERBASIC，XBASIC，BASIC PLUS等等。直到1978年才由美國國家標準局（ANSI）爲BASIC發展了一套工業標準化的BASIC，但也只是一個基準（Minimal）BASIC而已。除了基準BASIC外，今天仍然有許多不同版本的BASIC語言，它們之間都很類似，但也都各自有它們不同的特點。

在微電腦的領域內，使用最廣的BASIC版本是由MICROSOFT公司所寫的MICROSOFT BASIC，這些BASIC可在不同的微電腦上使用，但用不同的方法執行。IBM個人電腦內有3種BASIC語言（分別爲卡帶BASIC，磁碟BASIC，及BASICA），爲3種不同層次的MBASIC語言。所以這本書內我們雖以IBM PC爲主，但因MBASIC的互通性，故在其它的機器上也可使用如常，同時我們也盡量的使用所有BASIC版本內所共有的特性，以使本書具有較廣的涵蓋面。使用MICROSOFT BASIC的系統計有CP／M作業系統內的MBASIC或BASIC-80，IBM個人電腦之BASIC，

TRS-80之LEVEL Ⅱ BASIC，ALTARI系統BASIC，及TI BASIC等。同時請注意，一個良好易懂的程式應能放在不同的計算機上執行，或是稍加修正便可適用於其它的計算機上。

應該如何使用BASIC語言

平實的編寫程式

要想寫一個較長較複雜的程式，就必須接受平實的程式技術（Conservation Programming techniques），使程式內的錯誤所在能輕易的找出。（每個程式員，再有經驗也仍然會犯錯，你我偕如此）。這就是說應儘量避免使用某些BASIC版本內過於花俏的指令，除非程式已通過測試過程，並確定程式已能正確的執行我們所想要作的事後再加以考量。雖然如此，仍需斟酌是否要使用這些「特技」，尤其是關於一些列印及圖形的輸出指令，在它種計算機上可能根本不存在。最好把這類指令從程式中刪除，除非你覺得你是非用它不可。

我們曾發現並非所有的BASIC軟體特點都能像廠商所提供的參考手册內所描述的那樣執行。其中有些是語意上的誤解。所以在寫程式的技巧上愈保守或愈平實，則陷入程式內的陷井的機會也就愈小。這一章即將討論這種方式——即“平實的程式技術”。

編寫平實的程式，它的理由之一是程式可適用在不同的計算機上或可輕易的轉換過去。你可能會反問：「程式的易傳性（Portability）干我何事？」最主要的理由是你免不了會和朋友交換程式使用，但你所有的朋友都使用相同的計算機嗎？除非如此，否則他們若不修改你的程式的話將無法正常使用。易傳的程式技術將使修改量減低到最小。

易傳性對您自身的方便也是很重要的因素之一。你目前使用某一種計算機，可能在將來會變更使用另一種，或許使用更好的系統。爲了使您今天寫的程式，也能用在將來的計算機上，編寫程式時應多使用「平實的程式技術」。

使用平實的程式技術以使程式：
(1)易於分析及找到錯誤的所在。
(2)避免軟體內的陷井。
(3)强化程式的易傳性。

寫個易讀的程式

先參閱一下本書內所列舉的程式，你將發現它們讀起來淺白通順，又容易瞭解。因爲程式和敍述（Statement）都以簡單而又直接的BASIC敍述寫成，拋棄了玩弄技巧的花俏方法及特殊的BASIC語法。就好像這些敍述寫出來是爲了給讀者們閱讀用的，而與電腦無關一般。

寫一個易讀的程式需要於事前先思索，計劃出一個合於邏輯的流程，同時使用一些特殊的格式，使程式列印（List）後清爽悅目。若你以寫程式爲業，可能會受到老板個人的程式風格的限制。但若純爲個人的興趣而編寫程式的話，送給別人一個清爽，易讀的程式，除了那份得意與驕傲之外，將來若需更改或除錯時也會方便不少。

一個具有易讀性（Readable）格式的程式，在它程式的內部就可提供了部分的文件說明。程式的自我文件說明（Self-documentation）不單只是爲了閱讀，它同時也提供了讀者及使用者充分的訊息來說明程式的功能。這種格式並不與「結構化程式」完全相同，但我們借用了許多結構化程式中所强調的特點。我們使用模組（Module）的方式來組成一個具有組識格式的程式，每個模組具有一個主要的功能或是程式中的一件工作。我們也介紹了一些長久已來便爲大家所接受的優良程式技巧，但這些技巧在最近幾年有被人遺忘的現象。大多數的建議都無法節省記憶體空間，或是縮短執行時間。但是我們寧願多强調程式的易讀性而犧牲記憶體空間。在本章的後部，我們將看到如何縮短程式及增加執行速率的過程。但模組化的程式通常是較利於程式的執行，也較能夠

有效的將你的思考過程傳達給讀者。

上而下的組織

當我們在規劃一個程式時，可先考慮程式中幾個主要功能，這些功能可能是本次功能項目的部分或全部。

資料輸入（Data Entry）
資料分析（Data Analysis）
計算（Computation）
檔案更新（File update）
編輯（Editing）
產生報表（Report Generation）

所謂的模組程序是先將程式劃分成幾個模組，每個模組包含上述功能中的一個。程式中的流程從第一個模組進入第二個模組，而後再流進下一個模組。這種"上而下的組織"使得我們可以進行程式中流程的追踪。每個模組又可再分割爲數個程式塊（Block），每個程式塊各擔任一項程序（Procedure）或計算。程式塊的大小可由程式員視所欲完成的工作決定。程式塊的形態也常因人及程式而異。

寫程式應使用模組化的格式及上而下的組織方法

REMARK敍述

我們可利用REMARK敍述或空白的敍述列來使程式內的模組及程式塊彼此分離。但因有許多BASIC 版本在列印時會將空白的列編號列消除，所以先試試你的BASIC 版本在列印（LIST）時如何處理它的空白列。通常爲了程式的

易讀性，可隨意的增用 Remark 敘述，但也不要使用的太過份。一列空白列大多可用省略號“'”代替REM，或者只在列編號後加一個冒號“:”。也可用列編號後跟者一個REM。試試你的計算機，看那種情形可以存在。

```
100 REMARK         DATA ENTRY MODULE
110 REMARK ***** READ DATA FROM DATA STATEMENT 8000-9090
120 '
130 :
140 REM
.
.
.
190
200 REMARK  COMPUTATION MODULE
```

在程式中對一個模組、程式塊或副程式，可用一個註解的REMARK來開頭，及用空白列或空白的REMARK列來表示結束，（見上面列號190）。

在程式中使用REM或REMARK，必須前後一貫，這樣可增加程式的易讀性。使用省略號“'”或冒號“:”也要前後一致。有些作者使用＊＊＊（上例110列）區別出含有註解的REMARK。也有些人在REMARK後放置4到6個空位，再加上註解（列編號100）。兩種格式都能有效的在BASIC敘述中區分出REMARK註解。

我們也能使用多重敘述句的方式，將REMARK敘述與其他的BASIC敘述放置在同一列中，但是要注意到REMARK敘述必需是那一列的最後一道敘述。這種“同列”的註解可用來說明某一特殊的敘述是作什麼用的。通常這種同列的使用方式都用相當的空格把註解與BASIC敘述分開。

```
220 LET C(X) = C(X) + U: REM***COUNT UNITS IN C ARRAY
240 LET T(X) = T(X) + C(X): REM***INCREASE TOTAL ARRAY
```

我們確實希望能充分的使用REMARK來分隔模組與程式塊，同時也用REMARK來描述程式在做什麼事。但是別過份的濫用，像LET C＝A＋B就不必用REMARK來解釋。REM應該用來增加訊息，而不是用來再重覆已