

Apple II BASIC

王邦義校閱 柯建平編著

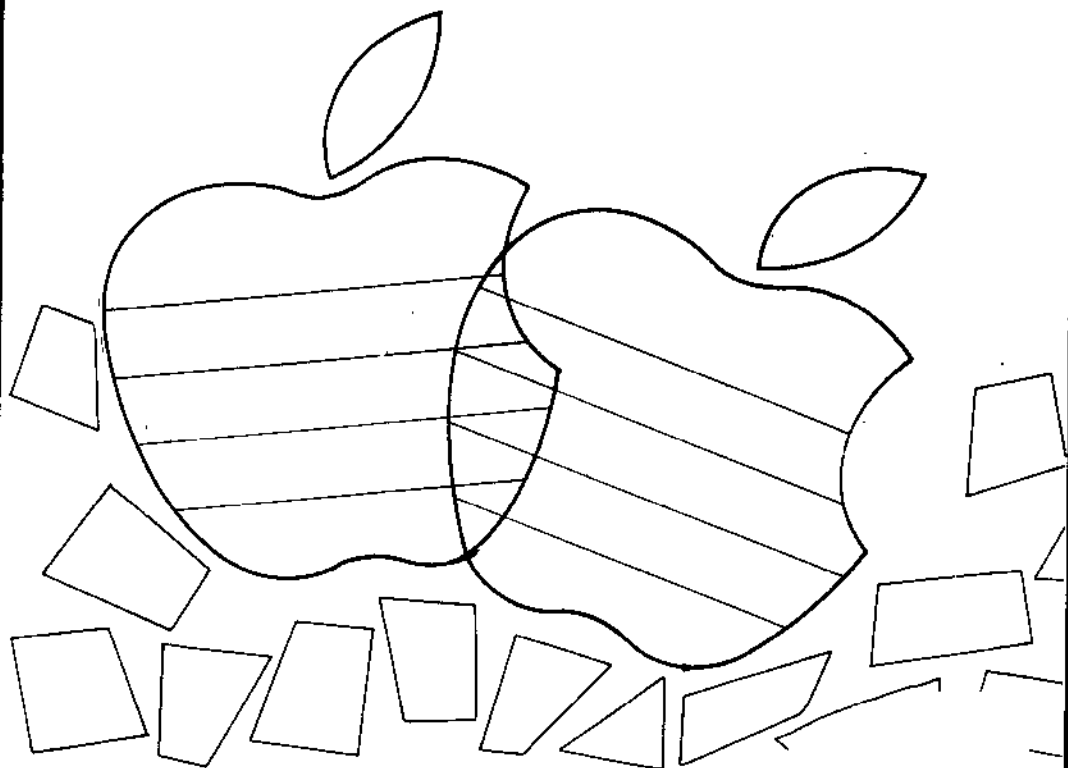


全華科技圖書股份有限公司 印行

Apple II BASIC

程式設計基礎

王邦義校閱 柯建平編著



全華科技圖書股份有限公司 印行



全華圖書 版權所有 翻印必究

局版台業字第0223號 法律顧問：陳培豪律師

Apple II BASIC 程式設計基礎

王邦義校閱 柯建平編著

出版者 全華科技圖書股份有限公司
北市龍江路76巷20-2號
電話：581-1300・541-5342
581-1362・581-1347
郵撥帳號：100836

發行人 陳 本 源
印刷者 佳怡彩色印刷廠
定價 新臺幣 150 元
再版 中華民國73年2月

序

爲讀者尋找一系列的操作例子與作業，使您能夠在電腦上實際的執行，於是您逐漸地“瞭解”了 BASIC，而這樣的學習方式猶如在還沒有進入學校以前學習我們的自然語言一樣。當您採用本書配合至少 40 個鐘點以上的實際作業之後，相信您將能夠與您的電腦以 BASIC 語言交談，換句話說，奠定 BASIC 程式設計的基礎與能力，是本書的目的。

本書以自學輔導的原則，而效以單元教學法的步驟編排，全書分爲電腦概念與操機、指令的使用及演練、程式設計規劃及範例參考三大部份。每一部份視進度的需要分爲若干個小單元，其中需要操作與演練的單元，其編排的方法與目的是：

1. 研究例：在每一單元的開始，均有一簡短、實用的研究例，用以引起學習的興趣或提示在本單元中將要學習的內容。
2. 討論與研究：在此活動中，舉例說明了研究例中未曾出現的敘述或符號，並提出相關的問題，希望讀者證實說明的正確性以及體會出自我研習的方法。
3. 複習與補充：將重要的討論結果歸納一遍，並且補充可能不容易明瞭或沒有提及的知識。
4. 練習題：每一單元的最後都有重點式的練習題，做爲此單元的學習目標，每一個練習都有解答，讀者可以自行比較與評鑑學習結果。

在此必須說明的，每一種廠牌的電腦所提供的 BASIC 指令不一定是完全相同的，但我們不得不確定一部可以實習的機種，如果 APPLE II 是您的選擇，那麼本書討論中所提供的研究問題，正可以幫助您以同樣的方法去瞭解其它廠牌類似的指令。

本書在一次偶然的機會裡，由李華敏老師提出一些 BASIC 電腦語言教學的觀點，再加上其他同仁們的鼓勵，終於在一面編寫一面試教驗證中完成。在此，特別感謝李華敏老師及熱心提供意見與指正的同仁們，同時也向全華書局不厭其煩的編輯、打字同仁們表示感激！

柯 建 平 謹識

中華民國七十二年八月

單元目錄

單元0 認識電腦及電腦語言

0-1 電腦的發展簡史	2
0-2 電腦系統的分類	2
0-3 電腦的硬體組合	3
0-4 電腦的軟體與分類	4
0-5 BASIC 在高階語言中的地位與用途	6

單元一 程式的編輯、執行與存取

1-1 程式的術語	8
1-2 程式的鍵入	9
1-3 程式的表列、修改與插入、刪除	10
1-4 程式的執行，中斷與恢復	12
1-5 程式的儲存與取用	13

單元二 循序式的流程與程式

2-1 輸入與輸出的敘述 INPUT, PRINT	16
2-2 輸入、輸出的標示	23
2-3 永久性資料的安排與讀取 READ.....DATA	30
2-4 螢光幕的控制命令與單字元輸入命令	37

單元三 (非)條件分歧流程與程式

3-1 非條件分歧GOTO敘述	48
3-2 非條件分歧GOSUB敘述	56

3-3	條件分歧之敘述 IF THEN	65
3-4	條件計值分歧之敘述	78

單元四 計數環路

4-1	單一環路與 FOR NEXT 敘述	88
4-2	巢式環路	96

單元五 數系的轉換與字串的取捨

5-1	十六進制轉換成十進制	106
5-2	十進制轉換成十六進制	113

單元六 低解像度畫面之應用

6-1	靜態之低解像度繪圖	122
6-2	動態之低解像度繪圖	132

單元七 高解像度畫面之應用

7-1	高解像度繪圖	142
7-2	位元圖樣的製作	148

單元八 程式的編寫與除錯

8-1	程式應有的特性	160
8-2	程式的規劃與編寫	166
8-3	程式的測試與除錯	169
8-4	結語	170

單元九 程式範例

9-1	雙骰兒賭	172
9-2	星期幾	175
9-3	調和數	178
9-4	等比級數	180
9-5	二次方程式	184
9-6	電阻性衰減器設計	187
9-7	兩個未知數之聯立方程式	189
9-8	計時加法測驗	191
9-9	萬年曆	198
9-10	成績統計圖	202
附錄○	APPLE BASIC 常用指令格式	207
附錄一	流程圖的圖形與功能	223
附錄二	數學函數與 BASIC 指令對照表	226
附錄三	APPLE BASIC 中之保留字	227
附錄四	錯誤訊息表	228

指令目錄

AND	68	LEFT\$	109
ASC	89	LEN	97
CONT	13	LET	17
CHR\$	89	LIST	10
COLOR	124	LOAD	14
DATA	32	MID\$	109
DEL	12	NEW	12
DIM	144	NEXT	90
END	20	NOT	70
FOR/NEXT	90	NORMAL	97
GET	39	ON/GOTO	79
GOTO	50	ON/GOSUB	80
GOSUB	59	ONERR GOTO	81
GR	133	OR	69
HGR	143	PEEK	134
HCOLOR	144	POKE	134
HLIN	123	PLOT	133
HOME	38	PRINT	19
HPLOT	144	READ	32
HTAB	39	REM	49
IF/THEN	66	RESUME	33
INPUT	19	RESTORE	33
INT	18	RETURN	80
INVERSE	97	RIGHT\$	110

RND	60
RUN	13
SAVE	14
SCRN	135
SGN	125
SPC	41
SPEED	89
SQR	21

STOP	80
STR\$	97
TAB	31
TEXT	124
VAL	97
VLIN	124
VTAB	38

程式例目錄

1. 輸入單價、數量值，然後求出總金額	16
2. 輸入國文（A），英文（B），數學（C）三科之成績，求出平均（V）及總分（T）	23
3. 將單位為公尺（m）的長度換算為公分（cm），英吋（in）和英尺（ft）	30
4. 依文法結構組句練習	37
5. 算術平均值之計算	48
6. 減法練習	56
7. 輸入A，B兩數，判斷其大小，並以“Y”表示繼續做這工作，輸入其它的鍵就停止	65
8. 輸入某班學生某科的成績，換算成A，B，C，D四個等級（A=88~100分，B=74~87分，C=60~73分，D=60分以下）	78
9. 將電腦所用之文字、符號以數字全部印出，並標註所代表之ASC 11碼	88
10. 列出九九乘法表	96
11. 輸入一十六進制之數，將其轉換成十進制之數印出	106
12. 輸入一十進制之數，將其轉換成十六進制之數印出	113
13. 繪出彩色條紋	122
14. 雷射光打U.F.O遊戲	132
15. 在螢幕上繪一最大的橢圓	142
16. 在高解像度畫面上顯示“BITS PATTERN”的字樣	148

單元



認識電腦及 電腦語言

0-1 電腦的發展簡史

從發明第一部電腦到現在約經過三十年的時間，在這期間，電腦的變化與進步確實十分驚人。其演變的過程，大至可以劃分為真空管時代，電晶體時代，積體電路時代，及大型積體電路時代，從下表可以看出演變中進步的情形。

代 別 項 目	第 一 代	第 二 代	第 三 代		第 四 代	
發 展 年 代	1952 ~ 1957	1958 ~ 1963	1964 ~ 1970		1971 ~	
電 子 元 件	真 空 管 Vacum Tube	電 晶 體 Transistor	IC SSI MSI		IC LSI VLSI	
單一包裝中相同功能之電子元件數	1	1	1 { 10	10 { 100	100 { 10,000	10,000 { 250,000
每秒鐘之加算次數(5位數)	1,000	10,000	25,000	40,000	60,000	100,000
單位成本之作業功能	1	10	100	1,000	10,000	100,000

IC : 積體電路

SSI : 小型積體電路

MSI : 中型積體電路

LSI : 大型積體電路

VLSI : 超大型積體電路

0-2 電腦系統的分類

1. 依處理之資料信號形態區分時

- (1) 數位電腦 (Digital Computer)
- (2) 類比電腦 (Analog Computer)
- (3) 混合電腦 (Hybrid Computer)

2. 依使用之對象範圍區分時

- (1) 汎 (通) 用性電腦 (General Purpose Computer)

(2) 特殊用途電腦 (Special Purpose Computer)

3. 依功能價格之高低區分時

- (1) 大型電腦 (Large Scale Computer)
- (2) 中型電腦 (Medium Scale Computer)
- (3) 小型電腦 (Small Scale Computer)
- (4) 迷你電腦 (Mini-Computer)
- (5) 微型電腦 (Micro Computer)

通常在無特別說明時，多指數位電腦或汎用性電腦。

0-3 電腦的硬體組合

電腦 (Computer) 是各型計算機的一個俗稱，它由以下五大單元所組成：

- (1) 算術邏輯單元 (arithmetic & logic unit)
- (2) 控制單元 (control unit)
- (3) 記憶單元 (memory unit)
- (4) 輸出單元 (output unit)
- (5) 輸入單元 (input unit)

算術邏輯單元 (ALU) 在整部電腦中負責計算、分析資料的工作，相當於人類的大腦。

控制單元 相當於人類的中樞神經，它有特定的傳輸線通往電腦的其他部門，控制電腦每一個時刻 (通常以微秒 10^{-6} S 為單位) 應有之動作。

記憶單元 可分為主記憶體 (main memory) 以及輔記憶體 (auxiliary memory) 兩部份，猶如人類對事務之處理一樣，沒有記憶力的頭腦 (主記憶體) 是不可能有所作為的。人類將大量的資料記載在記事本、書籍……等可以查閱之地方，這種輔助記憶之方式，正像是電腦將一些不急於處理之資料存入了磁碟機、磁帶機……等輔助記憶體之中。

輸出單元 是電腦唯一可以表現自己的部門，它可能是一個螢光幕 (CRT) 一個可以寫字繪圖的列表機，可以發出聲響之喇叭，或者只是十字路口上的一盞紅綠燈，依其功能的需要可多可少。

輸入單元 是電腦的感觀，借以與人類溝通或得知週邊反應的一個部門，它可能是鍵盤、讀卡機、磁帶機或者僅僅是一根可以探知溫度變化的探測器。

0-4 電腦的軟體與分類

前面所談及的電腦之各個單元都屬於電腦的硬體 (hardware)，硬體猶如一個人的身軀，欲使其有良好的作為，應附予其良好的教育，教育的工作能夠讓一個人在特定狀態下有正確的反應。對電腦而言，此教育的工作，正是替電腦安排一些軟體 (software)。說得通俗些，就是寫一些程式 (program) 給電腦去執行，由於程式沒有形狀、體積，它可以貯存於電腦的主記憶體或輔助記憶體中，故稱其為電腦的軟體。

寫一個程式去教導電腦如何工作。此間，人類就必須克服與電腦溝通的語言，所謂程式也正是使用某種電腦語言去安排一大串電腦能夠辨別的命令 (command)，而其使用之語言可分為以下三類：

1. 機器語言 (machine language)

是數位計算機 (Digital computer) 唯一能夠直接辨別的語言。此種語言在電腦中對話時，使用的方法是利用「正」「反」兩個信號，依照電腦能夠辨認的秩序規則變化，以達到溝通的目的。人類則以二進制中的“1”與“0”兩個數字表示機器語言中的「正」與「反」。下面是一段機器語言寫的程式，我們稱為目標程式 (object program)，它的目的是要兩數相加後，將其結果置於電腦記憶區位置 0200 處。

二進制編碼的機器語言	十六進制表示的機器語言
10101001	A9
00010000	10
00011000	18
01101001	69
00100000	20
10001101	8D
00000000	00
00000010	02

以上二進制編碼的機器語言是很難看懂的，即使我們以十六進制編碼將它縮短只剩下兩位數字的編碼，仍然必須經過查表才容易瞭解其動作的內容。(二進制、十六進制數系的轉換，請參閱單元五)

2. 組合語言(assembly language)

是人類以縮寫的文字符號來表示機器語言的編碼，如下所示：

以十六進制表 示之機器語言	組合語言	說 明
A9	LDA # \$ 10	；將十六進制的數 10（為十進制的 16
10		）載入計算機 ALU 中的一個加算器 A
18	CLC	；清除進位標記
69	ADC # \$ 20	；將 A 中之值與十六進制中之 20（為
20		十進制的 32）相加
80	STA \$ 0200	；存入記憶位置 0200（十六進制）
00		
02		

從上面的轉換說明中，我們可以揣測出 LDA 相當於英文 *LOAD ACCUMULATOR* 的縮寫，\$ 表示十六進制，# 則用以區分 \$ 10 是一個運算的資料，而不是從位置 \$ 10 去取一個資料。CLC 是 *CLEAR CARRY* 的縮寫，ADC 則是 *ADD ACCUMULATOR AND CARRY* 的意思，STA 則是 *STORE ACCUMULATOR TO ...* 的意思。

當電腦從程式中依序讀取到這些組合語言所用的文字符號以後，它必需經過組合語言翻譯器，轉換成機器語言，電腦才能執行。此過程稱為組合（assembly）。

3. 高階語言(high-level language)

可算是平易近人的語言，例如前面用機器語言以及組合語言所做的事，在使用 BASIC 高階語言來寫程式只需用下面一道命令表示：

LET RESULT = 16 + 32

多簡單啊！大多數的人在不懂得電腦系統結構以及動作原理或是數系轉換以前，已經可以用高階語言寫出很實用的程式。這份功勞要感謝電腦本身所儲備的一個翻譯者，它是一個特殊的程式，依其方式的不同分別稱為翻譯程式（interpreter program）及編輯程式（compiler program）。翻譯程式的翻譯方法是看完使用者程式一句紋

述後，立即翻譯成機器語言，並且執行出其結果。例如電腦在執行 *LET RESULT = 16 + 32* 這句敘述時，翻譯程式會告訴電腦去執行前面所提及的兩數相加的機器語言程式，而後再回來翻譯並執行下個敘述，直到程式中沒有敘述為止。而編輯程式的翻譯法是將使用者程式每一個敘述全部翻譯成機器語言後，才可以執行。若論其優劣，前者對電腦而言，等於是使用者說一句，電腦聽一句，所以執行整個程式的時間較長，但不需要留出目標程式（機器語言）的記憶區。至於後者，則等於是用電腦自身的語言（機器語言）作整篇的翻譯工作，所以電腦在執行目標程式時，必須等待一段翻譯的手續及時間，並且必須空出一段記憶區專門記錄翻譯後的目標程式，一旦翻譯完後，電腦執行工作，就如閱讀自己本國的語文，速度就比前者（*interpreter program*）快得多了。

0-5 BASIC 在高階語言中的地位與用途

高階語言的種類很多，依其特定的用途區分，使用較廣泛的有 *FORTRAN*，*COBOL*，*BASIC* 以及 *PASCAL*。*FORTRAN* 為一翻譯語言（*FORmula TRANslation*），適用於科學工程的應用。*COBOL* 為商用的語言（*COMMON Business Oriented Language*），適於商業統計及資料處理。*BASIC* 則為通用符號指令語言碼（*Beginner's Allpurpose Symbolic Instruction Code*）所構成的一種電腦語言，語法與英文接近，容易記憶，是一種交談性的語言，故常用於教學、商業以及娛樂方面。*PASCAL* 則是綜合以上各種語言的特色發展出來的一種結構式語言，語法較嚴謹，但應用的範圍較廣。

由以上簡單介紹中，我們不難比較出 *BASIC* 高階語言雖不是為某種專業用途而設，但事實上一個熟練的程式設計者，已經可使用 *BASIC* 解決廣泛的問題，故目前各公司推出的各種電腦系統均擁有使用 *BASIC* 語言的能力。尤其是逐漸走入家庭的微電腦，能夠使用 *BASIC* 電腦語言更是必備的能力。

單元



程式的編輯、
執行與存取