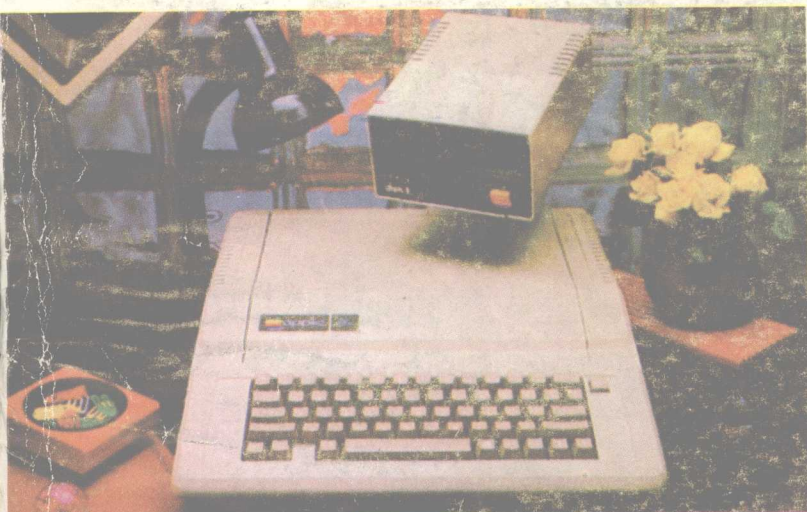


# APPLE II

pascal 入门



北方电脑公司资料组

# Apple Pascal 入門

北方电脑公司信息资料部

## 譯者序

常言“好的開始是成功的一半”，這意謂著起步的重要性；開始不只是一件工作成敗的決定要素，同時也是研究一門學問的樞紐，而本書正是為想開始學習PASCAL的人而寫的，因此，它的結構不但言簡而意賅，平易近人，看起來無吃力、不清楚之感，而且更重要的就是，它極盡一位老師之責：循循善誘，使您有不知不覺而竟然學會了的感覺，套句作者的話，要學習程式語言為什麼不選擇易於開始的PASCAL呢？那麼我們似乎應該這樣地說，要學習PASCAL為什麼不選擇這本易於會入的書呢？

另外，本書還具有一個很重要的特色，一般書都重理論，往往偏於談理而廢事，可是本書不但能以最簡單的方法把理論托出，而且更能配合實際的操作作業，可說理事雙收，一箭雙鵰，等您學完了PASCAL以後，那麼您對於Apple的一些操作也學會了，您說妙不妙？

最後我很感謝洗鏡光先生的鼓勵和指導，同時也希望這本書能帶給您無量的利益和智慧。

# 原 序

我們寫這本書一直所持的目的就是爲您，爲讀者找一系列的操作作業，使您能夠在您自己的電腦上實際的執行，於是您就逐漸地“瞭解”了 PASCAL，而這樣的學習方式就像您還沒有進學校以前在學習您的自然語言一樣的，我們假設您一開始時只懂得很少的電腦或電腦語言，但是我們希望，在您完成這 14 章和 30 到 40 個鐘點的實際作業以後，您將能具有幾乎知道所有字眼、文法規則、和構成 PASCAL 意義的知識了。

如果您正在學校或大學中學習 PASCAL 的話，那麼在這本書中的那些電腦操作作業將會給您一些實際的經驗，於是您對於在課堂上所學習的那些規則和抽象的理論也就更能有個實質的瞭解了；另外，如果您不是個在學生，同時您也不能或者不喜歡參加那種正式的 PASCAL 訓練，那麼本書將可以讓您在自己的步速和控制下來發展這套語言的實質知識。

我們認爲這類的書必須不能有一般教科書的缺點，因爲大部份的學習都是在當您坐在電腦前面執行一些實際操作作業時得來的，而這種情況很可能是您單獨一個人或者離任何專家的幫忙有一段距離，因此，這種書基本上應該字眼要清楚，程式字義要完整，同時基本的策略要能有效的執行。

在寫書的過程中，我們最感幸運的就是能夠找到一打左右的人，他們能夠放棄他們以前學習 PASCAL 的想法，而接收使用（並且評論）本書的早期草稿，他們在教書上，寫教科書上，程式製作上，清晰地思考上，以及在整頓那些雜亂的句子上提供了一些高度的發展技巧，Alan Portis 是一個熱心支持和提供好主意的泉源，Tim Aaronson，他教導年青人程式製作已經有好幾年了，因此他對於那些微妙的錯誤觀念和容易導致疏忽的地方特別敏感；Victor Jackson 對於那些矛盾和有毛病的程式特別具有慧眼，Martha Luehrmann，她對於那些少了符號的敘述和說明的抱怨，辯護是相當固執的，而在 Apple 電腦出版部門工作的 Elizabeth Weal，她志

願為我們那粗糙平淡的散文提供她長於鑑定的眼力和改正的技巧，於是我們就欠了她一筆特殊的債，至於其它致力於早期草稿工作的有 Arthur Kessner, Harold Peters, Robert Fuller, Dean Zollman, 和 Marvin Marcus。

我們很感謝文件處理系統的使用，因此我們就可以專心於那些批評和建議的改變，而很快地再把它修正過來，於是有幾章從一開始到我們認為滿意為止，前前後後共重寫了六次之多。至於本書中的程式片斷都是在電腦的控制下而從磁碟檔案中產生出來的，而且它們都是事先被執行過的，最後，這本書是由 Compugraphic 這個機器從最後的文件處理檔案中把它排版出來的，我們希望這個能消除一些打字上的錯誤，如果還有錯誤的話，我想您將會原諒我們的。

Arthur Luehrman

Herbert Peckham

---

# 介 紹

---

---

如果您已經擁有了這本書並且得接近 Apple II 電腦的話，那麼您可能很渴望地想開始來使用 Apple PASCAL 了，在這種情況下，您應該把這一章介紹的最大部份內容忽略過去，而直接跳到最後“如何使用這本書”那一節去。

另一方面，如果您是以顧客的心理——想看看這本書是否適合您，PASCAL 是否適合您，學習使用電腦是否適合您的話——那麼花一點時間在這章介紹上是值得的，我們在這一章上的目的就是在說明一點有關本書的設計和目的，回答“爲什麼需要 PASCAL 呢？”這個問題，以及簡略地說明爲什麼學習電腦是任何能讀能寫的人所值得努力的。

## 爲什麼要有這本書呢？

這實在是個很合理的問題，因爲關於 PASCAL 程式製作的書，包括這套語言作者所寫的那篇優美的論文在內大約有一打左右之多了，而這個世界真的還需要有另一本書嗎？

我們想這也沒有什麼好驚奇的，因爲：

- 據我們所知，並沒有任何一本書討論到電腦作業的細節問題，諸如鍵上的標籤，系統的編修程式，以及檔案系統等等，而對一個初學者來

說，這些東西實在是比其它任何東西還需要幫助的，因此，我們在這本書上的目的就是，教給您每一件爲了能在您的 Apple 電腦系統上寫，輸入，以及執行 PASCAL 程式所該知道的東西。

- 大部份的書都是那些在大學任教的電腦專家所寫的，因此，這些書的特色就是，它們所談到的數學，科學，以及商業應用都較適合於大學程度，可是同樣的那些程式製作觀念却可以使用其它更能爲大部份人所接受的應用範圍來說明，諸如：畫圖，文字內容的作業，或者產生音樂等等，而我們就是這樣地做的。
- 大部份的作者似乎都很關心語言的每一樣微小的細節以及巧妙的細微差別，好像那本書就是我們對於這個課目所要讀的最後一本書一樣，然而，我們却特別關心著使初學者能夠確實地做好開始，因此我們很少有那種不適當的字眼。
- 大部份存在的 PASCAL 書籍都有講義的味道，我們想，在這裏所需要的是的一本爲那些以做來學習的初學者而寫的書，而不是一本爲那些以聽和讀來學習的學者而寫的書，同時也是一本爲想要在自己步速下工作的人而寫的書。

我們在這本書上的目的就是引導您透過一系列使用 PASCAL 所得的經驗，以便達到跟您的電腦溝通的目的，而在溝通（以及不成功的溝通）的過程中，您也將會學到語言的基礎文法的，就像您在上學以前就已經學會了大部份的文法一樣。

請您不要嚐試著坐在舒服的椅子上來看這一本書，這種做法是不對的，務必請您走到放置 Apple 電腦的桌子旁邊坐下來，然後把這本書打開來放在您的膝上或者靠近電腦的地方，以便執行每一章所有的操作作業。

## 爲什麼需要 PASCAL 呢？

如果您已經熱中於 PASCAL 的話，那麼這個問題對您來說將不是個緊

急的問題，同時您也可能就會把這一節跳過去了，另一方面，如果您的回答是“PASCAL是什麼？”的話，那麼您最好把這一節看完了。

跟電腦溝通的意思就是，明確地陳述您的問題並且使用電腦語言把它寫成指令；在這幾年間人們已經創造了上百種的各種不同的電腦語言，而PASCAL就是其中的一種，這是一套相當新的語言，在1970年為瑞士電腦科學家Niklaus Wirth所定義的，至於其它比較流行而為人所知道的語言就是BASIC，FORTRAN，COBOL，ALGOL，PL/1，以及APL。

目前，知道如何使用BASIC語言來製作程式的人反而比其它任何的語言還要多了（由於在過去2年中幾乎賣掉了一百萬部使用BASIC語言的電腦），雖然如此，不過如果以使用的程式敘述列多寡來計算的話，COBOL還是比其它任何語言要多，這是一套商業應用的標準語言，可是大部份科學和工程上的程式還是使用FORTRAN語言來寫的，這是一套最古老的電腦語言，然而，應用數學家還是特別歡喜APL這套語言，至於電腦科學家，他們却把ALGOL當做在某些雜誌上發表程式的標準語言，而當IBM公司公佈了PL/1的時候，他們大力地宣傳這套語言將會很快地取代了FORTRAN和COBOL來處理任何的工作，甚至能做更多的事情。

因此，為什麼需要PASCAL呢？一個愚蠢的，但並不是完全不適當的答案就是“為什麼不需要呢？”不管擁護者和攻擊者之間的猛烈辯論，然而沒有一種共同的電腦語言會跟另一種語言相差太多的，每一種語言都大約有一百個左右的字眼和符號，每一種語言也都有一個嚴密的文法規則，好的定義以及沒有任何的例外，不過，任何一種語言的大部份“句子”都可以很容易地以及直接地轉換成其它語言的相似“句子”；所有的電腦語言大約都有個相同的“表示”範圍，而這些在一週內就可以學會的，因此，有經驗的電腦專家通常都是精通兩種或三種語言，因此，為什麼不需要PASCAL，或者任何其它語言呢？就像中國有句諺語“讓百花齊放…”。

我們並不是完全地不同情那種態度，在我們的觀點，無論所使用的語言是什麼，學習跟電腦溝通是件很重要的事，同時我們也不是受那些主張這種“好”語言將會逐出那種“不好”語言的人所感動，雖然所有的語言都是人類的創作，但是它們有它們自己受歡迎的期間和教化的身份，就像



## APPLE PASCAL 入門

英語的確並不優於世界語 ( Esperanto )，但是，當幾乎每一個人都使用世界語的時候，Shakespeare 和 Byron 却巧妙地使用英語說出了一些動人的事情來一樣。

實際上，我們想這些爭論應該是會被轉向的，因為電腦語言之間的不同實在太小了，那麼為什麼一開始時不選擇一種易於學習和適合於發展的語言呢？於是這就落在當初是為教電腦而設計的 BASIC 和 PASCAL 這兩種共同的語言上了，它們雖然在方法上有點不一樣，可是這兩種都是初學者最好的選擇，BASIC 是比較易於下手的，這部份是由於語言的設計，同時也是部份由於它的一般執行都是以相互溝通的方式在進行的，如果您使用 BASICS 磁碟片來“啓動”您的 Apple II 的話，那麼您就可以打入下面這列程式

```
PRINT 238 + 429
```

然後按一下 RETURN 鍵，於是馬上地您就會看到程式在執行，並且在螢光幕上印出“667”的結果來。

另一方面，PASCAL 下手是比較慢一點的，同時也需要具備更多語言和打入，編修，以及執行程式等電腦功能的知識的，就像上面 238 + 429 這個例子，在 PASCAL 上它就至少需要四列的長了：

```
PROGRAM ADDITION;  
BEGIN  
  WRITELN (238 + 429)  
END.
```

爲了打入這個程式，您必須進入編修程式 ( editor ) 中，然後您必須“離開編修程式”，最後再“執行這個程式”，當您在打入一個 BASIC 程式列的時候，如果不小心打錯了的話，那麼它就會告訴您，而您只要把那一列重新再打一次就可以了，可是在 PASCAL 中，您必須直到您執行這個程式的時候才會發現它的錯誤的，同時爲了要把它修正過來，那麼您就必

須再進入編修程式中，然後把它改正過來，接著離開編修程式，最後再把這個程式執行一次。

因此，BASIC 贏在易於教授，易於開始同時它的程式製作也易於測試和修改，然而，當 BASIC 程式變長了和需要被其它的人所修改的時候，那麼它的麻煩也就跟著來了，而這個却是 PASCAL 最大的長處之一：容易為讀者所瞭解；下面所示的那兩個程式例子就是模擬投擲骰子遊戲的 BASIC 和 PASCAL 的程式版本，請您不用去擔心它的細節問題，而您只要看看那一個版本看起來比較像個投擲骰子的程式就可以了。

#### Applesoft Basic 程 式

```

100 DEF FNR (X) = INT (1 + 6 * RND (X)) + INT (1 + 6 * RND (X))
110 LET R1 = FNR (1)
120 PRINT "YOU ROLLED A "; R1
130 IF R1 = 7 THEN 150
140 IF R1 <> 11 THEN 170
150 PRINT "YOU WIN"
160 GOTO 110
170 IF R1 = 2 THEN 200
180 IF R1 = 3 THEN 200
190 IF R1 <> 12 THEN 220
200 PRINT "YOU LOSE"
210 GOTO 110
220 LET R2 = FNR (1)
230 PRINT "NEXT ROLL IS "; R2
240 IF R2 = R1 THEN 150
250 IF R2 = 7 THEN 200
260 GOTO 220
270 END

```

## APPLE PASCAL 入門

### Apple Pascal 程 式

```
PROGRAM DICEGAME;  
  USES  
    APPLESTUFF;  
  CONST  
    HELLFREEZESOVER = FALSE;  
  VAR  
    DICE, POINT : INTEGER;  
  PROCEDURE ROLLEM (VAR TOSS : INTEGER);  
  BEGIN  
    TOSS := RANDOM MOD 6 + 1;  
    TOSS := TOSS + RANDOM MOD 6 + 1;  
    Writeln ('YOU ROLLED A ', TOSS)  
  END;  
BEGIN  
  REPEAT  
    ROLLEM (DICE);  
    CASE DICE OF  
      7, 11:  
        Writeln ('YOU WIN');  
      2, 3, 12:  
        Writeln ('YOU LOSE');  
      4, 5, 6, 8, 9, 10:  
        BEGIN  
          POINT := DICE;  
          REPEAT  
            ROLLEM (DICE)  
          UNTIL (DICE = POINT) OR (DICE = 7);  
          IF DICE = POINT THEN  
            Writeln ('YOU WIN')  
          ELSE  
            Writeln ('YOU LOSE')  
        END  
      END  
    UNTIL HELLFREEZESOVER  
  END.
```

除非您已經是個BASIC的專家以外，否則的話，那麼您對於第一個版本上的每一列到底是在做些什麼可能會花上很多的時間才能瞭解，相對的，在PASCAL的版本上，它的內容被分成各種不同的段，諸如，CONST

，VAR，和 PROCEDURE，另外，PASCAL也允許程式設計師使用突出縮進的結構來書寫程式，這就使得程式內容變得更清晰而易讀了，可是現在大部份的BASIC版本，對於那種突出縮進的程式列都把它當做錯誤來看待，並且著手把它從程式中移去。

當一個程式在大小和複雜度增加的時候，那麼BASIC的模糊不清和不可理解的問題却增加得很快，而PASCAL的好處也就更顯著了，根據經驗所得的結果，PASCAL程式設計師對於一個較長程式的修改和除錯往往比使用那些非結構式的語言諸如BASIC和FORTRAN等的人還要快幾倍。

為什麼PASCAL是一套最好而且第一個要學習的語言呢？這還有另一個其它的理由，一套比較新的語言對於其它的電腦語言來說，它往往能得到下面這種的利益，那也就是說，它常常是電腦科學家們經過長年累月思考的結晶，就像電腦科學家們對於為什麼那些以1960年初期的語言所寫的程式總是那麼難以證實是正確的時候，於是他們就著手於這項的研究，結果發現一套語言需要儘可能地讓一個程式以一段一段的方式來書寫，使人看了就知道它將要做些什麼工作，而PASCAL除了想產生一套應用的語言以外，而且還把這些研究的結果加進去了。

從一個很實際的意義來說，PASCAL是近代考慮某些電腦語言目標的工作典型，就像所有好的典型一樣，它正廣泛地被模倣著呢，最近美國國家標準局(ANSI)所定的FORTRAN標準也包含了一些跟PASCAL一樣的組織結構了，而在ANSI BASIC上的工作也正接近完成，它也包含了所有的這些結構，以及擴展變數名稱，和有名稱的程序。今日，在所有語言中最突出的，可說是Ada了，這套語言很快地成為所有想做跟美國防衛部有關工作的承包商所需要，以做為PASCAL的擴展。

因此，學習使用PASCAL來製作程式可說是變成熟悉那些不在PASCAL中而却出現在更舊或一些最新語言中的語言觀念的一個好辦法了，於是，如果您已經知道PASCAL的話，那麼下一種語言也就變成比較容易學習了。

雖然，它常常是解決工作的正確工具，然而，在一部電腦諸如Apple II上的很多工作，BASIC却反而是個正當的工具，因為它容易學會，能夠很快地給您診斷的訊息，以及易於再編修和測試——而這些恰是PASCAL

最脆弱的地方。雖然，往往那些簡單的問題都可以為那種快而髒，測試錯誤的方法所解決，但是這並不能證明它更適合於工作，當然，這種技巧在那種方法或許會失敗的時候是要知道的，這樣我們就可以避免它了，然而，根據我們實驗所得的作法就是，如果您的程式小得足以在您的螢光幕上顯示出來的話，那麼您就不妨使用 BASIC 語言來寫，但是，如果您的程式會更大或者可能會繼續變得更大的話，那麼您就得使用 PASCAL 來寫了。

### 為什麼需要電腦呢？

如果您已經知道電腦並且相信它是很重要的話，那麼這個問題對您來說，就無關緊要了，因此，您也就可以把這一節跳過去了，然而，如果您有所疑慮的話，那麼您就得把這一節看完了。

大部份的人都知道現代電腦的速度和正確性了，只要您一下命令，它就能把那些既長又乏味的計算很正確地做出來了，因此，它的足跡在工廠界，商店，學校，甚至家庭中不斷地在擴展著。今天，大部份的人可能都已經承認在他們的日常工作：代替工作，購買商品，接收賬單，使用支票或信用卡的付款，飛機或旅館的訂位，報時，電視機的開關，私人不重要信件的接收，自動電話撥號的使用，數字記錄音樂的收聽，影像磁碟記錄的收看，微波烤爐的時間和溫度的設定，或者現代家庭中自動調溫器的時間和溫度的設定，以及其它很多的事情中，電腦的確扮演了一個很重要而且決定性的角色，但是並不是每一個人對他所看到的都滿意，就像一個盲人依賴新科技一樣，尤其是當它，或者，常常是，它的操作員和程式設計師弄錯的時候，但是即使是那些吹毛求疵的人也承認電腦改變了今日的部份生活。

不管電腦是那樣的為人所週知，但是值得注意的就是，却很少人是受過電腦教育的使用者，因此，大部份的人對電腦的基本認識已經失掉了：亦即，電腦的行為是由人類所控制的；而一部電腦所做的就是它所該做的，因為我們人不但告訴它做些什麼，而且也告訴它如何去做，至於一個知

道如何使用電腦語言來表達思想的人，他也就有了一種新的談，寫，以及考慮一些觀念的方法了，同時在任何時候，當您發現一種考慮事情的新方法，或者一種敘述您的思想的新語言時，就把它們記下來以便做更仔細地探討，那麼您就會有一個解決問題的新工具了，關於電腦，就大體來說，這是最重要的了。

實際上，即使有那麼多人失掉了這種觀念也是沒什麼好奇怪的，因為我們正生活在很多方向的時代，就像印刷機被發明後的前面十年一樣，而那個時候正是除了一個小僧侶以外，其餘的幾乎每一個人都是文盲，當時人們也承認印刷術將使得商界和政府的交流以及記錄的保存變得更容易多了，而這個將會對於日常生活上帶來很大的影響，但是，幾乎沒有人，縱使在幾世紀以後，也沒有人會承認它的主要影響會出現在當整個國家都變成文明同時可以使用讀和寫來滿足個人的求知欲的時候呢！

廉價的個人電腦，它所呈現的現象也是跟印刷機所顯現的一樣，在大家的心裏，電腦能讀能寫的人仍然是屬於那個小僧侶的，他很辛苦地得到那些為門外漢所不能接近的技巧，另一方面，今日大部份受過教育的人都相信，無論電腦需要的是什麼，它們都可以使用英語告訴那個僧侶，接著由那個僧侶把它們的希望和禱告轉換成電腦語言，然後他們才跟這個不可思議的電腦溝通的。

當然，並不是這樣的，現在有很多的人正在發明易學，而且可以滿足個人跟電腦溝通和控制的方法，事實上，由最近的研究顯示出，大部份買電腦的主要理由並不是把它拿來做實際的使用，而只是用來學習跟它溝通以便成為電腦能讀能寫的人而已，當他們的技巧和知識成長了以後，於是他們就發明了一些預先都不能想像的用法。

現在我們應該為這一段做個結論的時候了，就跟學習讀，寫的理由一樣，您也應該要學習計算的，以便做做數學；因為它對您的智力是有益的。

## 如何使用這本書

就像以前我們所說的一樣，這本書使您有一連串的鍵盤操作作業，如果您把它當做一本傳統的教科書來看的話，那麼它可能會太簡單了一點，同時也可能會混淆您的閱讀，而您將會發現到書中所說的那些規則，說明，以及摘要，它們都是跟您的那些鍵盤上的實際試驗所得來的結果一樣的。

在本書中的前面五章或六章並沒有離開這種方法太遠，這是一個很好的策略，因此，如果您偶而進入您所不能控制的狀況時，請您不要驚奇，這並不會損害到您或者電腦的，當然，如果狀況愈來愈壞的話，那麼您常常可以把電源關掉，然後再重新啓動。

而我們在這裏的主要目的就是，使一個完全沒有經驗的人可以很成功地學習 PASCAL，因此，我們並不排除那種容易的方法，同時我們所給您的只是最基本的部份語言，所以您在學習 PASCAL 時很少會有這種現象，那就是您從它上面所學到的都是些其它語言逐字解釋的功能，取而代之的，我們很快地介紹給您程序和函數段，各種不同的型別，以及那些主要的 PASCAL 控制結構，而到了書本的後半部以後，您將會發現到一些程式設計師自己定義的資料型別，陣列，集合，以及記錄的例子的，這些都是些比較繁重的課題，不過每一個也都是建立在您以前的瞭解和事先的說明上，然後再加進新的，最後再做一些具體的試驗，因此，如果您的學習是循著這些操作作業穩定地前進的話，那麼我們相信您將可以很容易地接受每一個新的觀念或規則的，然而，即使您在學完第 8 章或第 9 章或第 10 章以後，如果您覺得對 PASCAL 已經足夠瞭解的話，其實在那個時候，您已經有了基本的瞭解了，那麼您就可以寫一些比較複雜的程式了。

至於電腦系統說明的問題，我們假設您有一部 Apple II 或者 Apple II Plus 電腦，一個語言系統，一個磁碟機，和版本 1.1 的 Apple PASCAL 磁碟片，如果您有兩個磁碟機的話，請您不要認為這本書不適合於您，雖然在這裏的所有作業只要使用一個磁碟機就可以了，但是您所學的每一樣東西也都可以在兩個磁碟機系統上執行的，另外，如果您的磁碟片是版本

1.0 ( 也叫做版本 11.1 的 UCSD PASCAL ) 的話，那您仍然可以使用這本書的，然而，在做每一章以前，您應該先參考附錄 H，於是您就可以知道版本 1.0 的磁碟片跟版本 1.1 的磁碟片有那些不一樣的了。



# 目 錄

介 紹	0-1
爲什麼要有這本書呢？	0-1
爲什麼需要 PASCAL 呢？	0-2
爲什麼需要電腦呢？	0-8
如何使用這本書	0-10
第一章 開 始	1-1
1-1 PASCAL 的啓動	1-4
1-2 COMMAND 提示列	1-7
1-3 參觀系統	1-8
1-4 比所看到的還要多	1-11
摘 要	1-12
練習題	1-14
第二章 打入程式—編修程式	2-1
本章的目的	2-1
2-1 準備工作	2-2
2-2 進入 EDITOR	2-4
2-3 打入新的內容	2-5
2-4 把插入的內容移到工作區域中	2-7
2-5 RESET 鍵	2-8
2-6 把工作區域的內容移到磁碟片上	2-10
2-7 編修的概觀	2-14