

FORTRAN IV

Programming Second Edition

微電腦程式初階

V. Thomas Dock 原著



進修電腦課程本書為上乘之選，
每章節均列練習操作與解答題。

微電腦程式初階

電腦資料供應社出版

微電腦程式初階

出版：電腦資料供應社

香港乍畏街38號

印刷：嶺南印刷公司

香港西安里4號

1982年8月版・版權所有

序 言

今天，大多數的電子計算機，都能提供許多程式語言使用，藉著這些語言，使用者可順利完成各種既定目標。FORTRAN便是其中之一。FORTRAN目前有許多不同版本，本教科書則採用最新且最普遍的版本—FORTRAN IV。

本書旨在以系統化的方式，提供讀者了解FORTRAN語言。前面幾章先討論FORTRAN的基本鍵語（key word），好讓讀者可立即開始撰寫程式。其餘各章所討論的鍵語，則讓讀者進一步了解FORTRAN IV語言的實用性，及其在撰寫程式上的變通性。書中所討論的FORTRAN IV語言是用於IBM 360與370系統上，它可與美國國家標準（簡稱ANS）FORTRAN語言通用，甚且涵蓋其上，例如它的數學函數部分包括在此語言內。因此，IBM FORTRAN IV語言可在下列編譯器（Compiler）上加以編譯及執行：IBM FORTRAN E、G、H，WATFOR，WATFID，以及ANS FORTRAN。

本書假想讀者未曾接觸此種語言。因此，在熟悉一些有關的指述（Statement）後，將從FORTRAN的基本組成開始討論。其中透過各種不同指述的介紹及其應用的例子，來揭開此種語言的應用潛力。此外，在每章的結尾都附有許多問題及練習題，以強調每章所討論的重點。其解答附於本書最後，使讀者能立即核對答案正確與否。

本人寫此教科書，要致謝的人太多，無法逐一提及，僅在此感謝北科羅拉多大學的Steve Teglovic博士，對本書的原稿有精闢的建議。

V. Thomas Dock

◀◀目次▶▶

第一章 電子計算機與FORTRAN IV之簡介 1

1-1 電子計算機定義.....	1
1-2 類比式與數位式電子計算機.....	2
1-3 FORTRAN IV 簡介.....	2
1-4 FORTRAN 程式範例.....	3
1-5 FORTRAN 程式編碼紙.....	3
1-6 FORTRAN 程式指述.....	3
1-7 註解指述.....	8
1-8 FORTRAN 的字元集.....	9
1-9 資料卡.....	9
1-10 工作控制語言卡.....	10
1-11 打卡機.....	11
1-12 原始程式轉換成目的程式.....	11

問題

第二章 FORTRAN程式指述之構成

第二章 FORTRAN 程式指述之構成..... 14

2-1 FORTRAN 字元集.....	14
2-2 FORTRAN 鍵語.....	17
2-3 數字.....	18
2-4 變數名稱.....	21
2-5 算術式.....	27
2-6 FORTRAN 指述之種類.....	28

問題

習題

第三章 輸入與輸出指述..... 30

3-1	READ指述.....	30
3-2	WRITE 指述.....	33
3-3	FORMAT 指述.....	35
3-4	其他可供選用之FORMAT.....	41
3-5	輸送架控制字元.....	41
3-6	格式碼.....	46
3-7	FORTRAN 習題	61

習題解答

問題

第四章 指定指述..... 67

4-1	算術指述.....	67
4-2	邏輯指述.....	75
4-3	關係運算元號.....	75
4-4	邏輯運算元號.....	77
4-5	執行層次.....	77
4-6	控制指述之簡介.....	78
4-7	無條件的GO TO 指述.....	78
4-8	STOP指述.....	79
4-9	END 指述.....	80
4-10	FORTRAN 程式範例.....	80

問題

習題

第五章 控制指述..... 89

5-1	系統流程圖與程式流程圖.....	89
5-2	分叉.....	89

5-3	循環	100
5-4	CONTINUE指述	106
5-5	兩張以上資料卡之讀入	108
	問題	
	習題	

第六章 陣列113

6-1	陣列之性質	113
6-2	陣列之建立	115
6-3	陣列之維度	118
6-4	下標與陣列元素	119
6-5	陣列元素填號之正常順序	120
6-6	所有陣列元素的資料搬移	122
6-7	特定陣列元素的資料搬移	124
6-8	部分陣列元素的資料搬移	125
	問題	
	習題	

第七章 副程式.....134

7-1	FORTTRAN 供應之副程式	134
7-2	FUNCTION 副程式	137
7-3	RETURN 指述	140
7-4	SUBROUTINE 副程式	142
7-5	多重入口之副程式	148
7-6	副程式名稱當作輻數之使用	150
7-7	指述函數	153
7-8	外顯規格指述	155
7-9	與副程式相關的程式指述	156
	問題	
	習題	

第八章 其他FORTRAN指述.....169

- 8-1 不同型態的輸入與輸出指述.....169
- 8-2 NAME LIST 指述.....171
- 8-3 其他規格碼.....175
- 8-4 PAUSE 指述.....177
- 8-5 DATA初值指述.....178
- 8-6 BLOCK DATA 指述.....179
- 8-7 IMPLICITT 指述.....181
- 8-8 COMPLEX 指述.....183
- 8-9 符號名稱的長度.....184
- 問題
- 習題

第九章 磁帶與磁碟指述.....193

- 9-1 磁帶.....193
- 9-2 磁碟.....200
- 問題
- 習題

- 附錄A IBM 29 卡片打孔機之簡介.....212
- 附錄B FORTRAN IV鍵語.....217
- 附錄C 程式除錯.....218
- 附錄D 以十進位指數表示的實數與整數.....246
- 附錄E 其他可供選用之格式.....253
- 附錄F 邏輯運算.....255
- 附錄G 兩疊以上資料組的讀法.....257

問題與練習題解答

第一章.....	262
第二章.....	264
第三章.....	267
第四章.....	268
第五章.....	271
第六章.....	273
第七章.....	275
第八章.....	280
第九章.....	283

第一章 電子計算機與 FORTRAN IV之簡介

對一位初學者而言，在進入 FORTRAN IV 程式語言主題之前，最好先具備一些電子計算機的基本常識。尤其是對執行此程式語言的機型有一番認識的話，必能提高往後學習的興趣。因此，下面我們先就“電子計算機”所涵蓋的意義及其兩種主要的型態，做一扼要的探討

1.1 電子計算機的定義

“電子計算機”這個名詞雖然適用於能夠做算術運算的任何機器。但是，一般所針對的機器乃是具有下列四種特性：

1. 電子式：能透過電子脈衝式的運轉，而非機械式的內部移動，來完成計算結果。
2. 內部儲存器 (internal storage)：有能力同時儲存程式中的指述 (statement) 和資料 (data)。如此才能使電子計算機以高速、連貫的方式執程式指述。
3. 內儲程式 (stored program)：能將一系列指述先放到內部儲存器中，並詳細指示出應做的運算以及執行的次序。
4. 程式執行中之修正：能改變程式指述執行時的路綫。例如分叉 (branch) 指述，可以根據一個以上的算術或邏輯運算所得的結果，或僅僅是內部儲存器某一位置的內含，來決定下一個該執行的指述。

簡單的說，電子計算機是一具電子機器，而具有內部儲存器，內儲程式指令 (instruction)，以及程式執行時修改指令路綫等等各種能力。

1.2 類比式與數位式電子計算機

電子計算機有兩種主要型態：數位式與類比式。數位電子計算機直接作用於非連續性（discrete）資料或符號所表示的數值上。它是以打卡孔，磁帶點，印刷字體等等所表示的數字（number），文字（letters），以及特殊字元（special character）作為輸入與輸出之形式。這種型態的電子計算機也就是我們普遍所謂的“電子計算機”一詞。

數位電子計算機通常用於商業上與科學上的資料處理。它能夠使資料的間接量與最後值達到各種不同的精確度，這就要看數位電子計算機本身的特性及其處理資料正確性而定。目前這一類型的電子計算機在商業上用得最廣泛。因此，除非另有聲明，本書所討論的電子計算機即指數位電子計算機。

類比電子計算機正好相對於數位電子計算機。因為它所度量的資料是連續性的電氣式物理單位，而非作用於數位性資料上，如果涉及到數位性資料，它將以間接方式取得。諸如壓力、溫度、電壓等物理量直接以連續函數測得。類比電子計算機的輸出通常用來修正機器之控制。例如，它可以調整活栓位置，來控制小管的流量。因此，它經常用來控制像鍊油或烘油等程序作業。數位電子計算機也可用於程序控制。為進行此種工作，類比資料必須轉換成數位形式，經過處理後，必須再將數位結果轉換回類比形式。綜上所述，數位電子計算機具有較高的精確性，而類比電子計算機則有較快的處理速度。

1.3 FORTRAN IV簡介

FORTRAN 這個專有名詞是由**FORM**ula **TRAN**slator兩者字首合併而來的。因此，它特別適合編寫算術計算或數值資料處理等應用程式。但這並非意味着 **FORTRAN** 不適於商業上的資料處理。此種語言不但可以處理商業上大量的數值資料，亦可以處理非數值的資料，如英文字母或特殊字元。接在 **FORTRAN** 之後的IV表

示此種語言目前的版數。

1.4 FORTRAN 程式範例

圖 1-1 是一個 FORTRAN 程式的範例。我們在此刻介紹程式，目的是想奠下討論 FORTRAN 語言的一些基礎。大體而言，此程式執行時，將資料卡片前 5 行所表示的數字後輸入裝置中轉移到電子計算機的記憶體中，並且儲存在變數名稱 (variable name) NUMBER 所指的記憶地址 (memory address) 中。然後再將此數字從記憶地址 NUMBER 中取出，並印在列印機新的一頁中的第一行上。而 “THE NUMBER READ FROM THE FIRST 5 PUNCH CARD COLUMNS IS ” 一句話立即印在此數之前。電子計算機將重複進行上述一系列動作，直到資料卡沒有為止。當它偵測出無資料卡片時，儲存程式的執行立即終止。

1.5 FORTRAN 程式編碼紙

一個 FORTRAN 程式是由 FORTRAN 指述 (statement) 以及可能有數個註解 (comment) 指述所構成。FORTRAN 程式編碼紙 (coding form) (見圖 1-1) 可以視為撰寫這些指述的最佳工具。撰寫完程式之後，其所包含的指述通常打在對應欄數的打孔卡片上。程式編碼紙上的每一列表示一張 80 欄的打孔卡片。

* 此時如果讀者尚未完全了解程式執行所涉及的概念，程序，以及規則，可以暫時不予理會。

1.6 FORTRAN 程式指述

前面我們提過，FORTRAN 程式編碼紙的每一列有 80 行 (column)，亦即表示 80 行的打孔卡片。每一列的字碼打在對應的卡片行上。當 FORTRAN 指述要進行編譯 (compiled) 時 (我們將在最後一節討論編譯處理)，它們將以打孔卡片相同的順序印在列印機上，並且構成所謂的原始程式清單 (source program listing)。(

IBM

FORTRAN Coding Form

ORDER FORM 4 (Rev. 6-67)
Printed in U.S.A.

PROGRAM	DATE	FUNCTION	NAME OF USER
PROGRAMMER			

LINE	FORTRAN STATEMENT	CHARACTER POSITION
1	WRITE (14,7344) NUMBER	
2	FORMAT (14,7344) NUMBER	
3	ATX 15	
4	THE NUMBER READ FROM THE FIRST 5 PUNCH CARD COLUMNS	
5	IS 99999	
6	END	

Figure 1-1. An Example of a FORTRAN Program.

FORTRAN Coding Form		NAME OF PROGRAM		DATE		REVISION		PAGE	
LINE	FORTRAN STATEMENT	FORTRAN STATEMENT	FORTRAN STATEMENT	FORTRAN STATEMENT	FORTRAN STATEMENT	FORTRAN STATEMENT	FORTRAN STATEMENT	FORTRAN STATEMENT	FORTRAN STATEMENT
1	RM EXEMPLE OF A FORTRAN PROGRAM								
2	READ (5,2) END=99) NUMBER								
3	FORMAT (754)								
4	WRITE (4,3) NUMBER								
5	FORMAT ('11' THE NUMBER READ FROM THE FIRST 5 PUNCH CARD COLUMNS								
6	AT=1 X 15)								
7	DO 10 I=1,15								
8	STOP								
9	END								

Figure 1-2. The Same FORTRAN Program as is Shown in Figure 1-1, except for Statement Numbers.

原始程式這個名稱將在最後一節討論。))

程式指述的號碼——如圖 1-1 所舉的例子，並非每一個程式指述都需要有號碼標示。只有所謂的 `FORMAT` 指述以及程式執行時欲轉往的指述，必須要有一指述號碼附加於左。這一類的指述將在第三章及第五章詳盡討論。下面兩段僅對其基本做一扼要說明。

`FORMAT` 指述是程式指述的一種，通常有一個或數個 `READ` 指述或 `WRITE` 指述牽涉到它。因此，電子計算機必須要有些方法來鑑別 `FORMAT` 指述。

正常情形下，電子計算機是按照指述本身在程式中的順序執行，而非按指述號碼的順序。也就是說，第一個程式指述是第一個被電子計算機執行的指述，第二個程式指述則是第二個被執行，等等。可是，如圖 1-1 所示，透過 `GO TO` 指述的使用，`FORTRAN` 語言可提供向前或向後轉移程式指述的方法。而電子計算機必須具有鑑別是前或是後指述的方法。因此，指述號碼就是用來鑑別特定的指述。

打孔卡片第 1 行到第 5 行——如 `FORTRAN` 程式編碼紙上所示，程式指述所對應打孔卡片的第 1 行到第 5 行只能包含某一特定指述的號碼。這個號碼可由數值 1 到 5 的五個數字組成（此 5 行只允許數字出現），接下去的指述號碼不必按大小順序給予。換句話說，圖 1-1 所示的 `FORTRAN` 指述可以寫成圖 1-2 所示的指述號碼，兩者程式指述的執行順序是相同的。但是，為了使打卡與檢查方便起見，將建議讀者無論如何應以圖 1-1 為例，按次序編寫指述號碼。

然而對上述的建議有一例外，那就是在有需要給予 `STOP` 指述一指述號碼時。其原因是在大多數 `FORTRAN` 程式中，讀者所使用的指數號碼最多不會超過 98 個，而且在正常情況下，你應給予 `STOP` 指述一指述號碼。由於這個事實，再加上號碼 99 既獨特又易記，所以常將此號碼配合 `STOP` 指述使用。

指述號碼可以在卡片 1 到 5 行內任一行開始及終止，號碼中的數字也不必兩兩相連，因為電子計算機並不鑑別居間的空白。因此下列指數號碼的打法：

(卡行) CC		1	2	3	4	5	6	7	8	9
		25								

與下列指述號碼的打法：

(卡行) CC	1	2	3	4	5	6	7	8	9
	2	5							

都可以接受。在上兩例中，電子計算機皆將指述號碼看成是 25。可是，此時仍然要建議讀者，爲了打卡及檢查方便起見，指述號碼最好從打孔卡第 1 行開始，如果含有兩個以上的數字，則將其打在連續的位置上。

程式中的指述號碼必須是唯一的。換句話說，兩個指述不能有相同的指述號碼。如此規定有其原因，如果許多程式指述有相同的指數號碼，那麼電子計算機就無法決定它要走向那一個指述。

打孔卡片第 6 行

亦如 FORTRAN 程式編碼紙所示，打孔卡片第 6 行僅用來表示程式指述是否連續。第一個程式指述的第 6 行必須是空白或零。如圖 1-1 與 1-2 所述，當程式指述超過 66 行而需要連續兩張或數張打孔卡片時，連續卡片的第 6 行必須爲一非空白或零的任何字母、數字、或特殊字元。其原因是因電子計算機把卡片第 6 行中的空白或零解釋爲非連續的程式指述（亦即指述已完全打在一張卡片上）。因此，對任何其他非空白或零的字元出現在卡片的第 6 行中，電子計算機就將此一程式指述解釋爲上一指述的延續。如果有必要，程式指述可以連續到 19 張卡片。

打孔卡片第 7 行到 72 行

如 FORTRAN 程式編碼紙所示，第 7 行到 72 行是用於 FORTRAN

指述本身。指述書寫不必非從第 7 行開始，亦不必非到 72 行結束。但是爲了打孔及檢查方便起見，建議讀者最好從第 7 欄開始書寫一新的指述。

除非有特殊情況（這些情況我們將在後面討論），程式指述之間的空白沒有必要保留。即使有的話，也不會影響指述的執行。因此，在一指述中，運用適當的空白，可以增加指述的可讀性。

打孔卡片第 73 行到 80 行

最後，FORTRAN 程式編碼紙的第 73 行到 80 行並不使用於 FORTRAN 指述中，因爲只有前面 72 行對 FORTRAN 編譯器有意義。73 到 80 行可以是空白，也可用作程式鑑別、卡片順序指引、以及其他任何目的。只要在這 8 行中打上你所需要的字碼即可。例如，若輸出裝置是一列印機，所打的字碼將與程式指述出現於同一列上。

1.7 註解指述

註解指述（comment statement）是用來說明或鑑別整個 FORTRAN 程式，如圖 1-1 與 1-2 所示，或程式中的某一特定部分。我們是用 C 字母來辨別註解指述，此字母必須打在卡片的第 1 行上。

註解指述與 FORTRAN 指述不同，它可以從卡片上第 2 行到 80 行中的任何 1 行開始打孔。如果需要連續註解幾張，則每張註解卡片的第 1 行仍然要打上英文字母 C，這一點也與 FORTRAN 指述不同。但對於空白的運用，以增加程式的可讀性，則與 FORTRAN 指述相同。

註解指述是一個非執行的（nonexecutable）指述。換言之，它既不爲 FORTRAN 編譯器處理（即翻譯成機器語言），也不影響 FORTRAN 程式的執行。但是它仍按照打孔卡片的順序隨同程式指述，依次印在原始程式清單上。

通常程式的開頭都會安置一個或數個註解指述，用來說明程式的大綱。有時候程式相當長，而包含數個可區別的部分，程式計劃員爲