

ANSI FORTRAN IV and FORTRAN 77

FORTRAN

程式設計與商業上應用

劉廼明 譯



松崗電腦圖書資料有限公司

ANSI FORTRAN IV and FORTRAN 77

FORTRAN

程式設計與商業上應用

劉廼明 譯

Nesa L'abbé Wu

松崗電腦圖書資料有限公司 印行

原著序

本書主要是為商科學生而寫，亦適用於其他科系有關電子計算機之基礎課程；對方修習工科與系統工程者而言亦頗為適用。

本書在取材與撰寫之方向，特別強調簡明、實用，因此亦為工商企業界人士有志探討計算機程式語言時，相當適切之參考書。

就FORTRAN程式設計之目的而言，本書已將有關FORTRAN通用規則作詳盡地研整、綜合並摘錄，個人以為此為FORTRAN程式設計之教學與應用之先行預備工作，流程圖為邏輯規劃時甚為有用之工具，因此作者將本書程式範例皆以流程圖與邏輯方式處理，並儘可能予以詳盡解釋。

本書內容最初係取材東密西根大學同事及本人授課講義與教材，後歷經三次大幅度修改與增補：包括標準FORTRAN 77 簡介、良好之程式設計技巧研討及附加資料之應用等，而新版本並強調良好程式設計技巧實作與結構化程式設計觀念，最後一章另有三個案研究，均有詳盡分析與討論。

學生與教師在使用本書前，應把握以下二原則；「勿徒耗時間於書上文字之閱讀」及「多以範例講授或表示」，特別應強調流程圖與邏輯之觀念。

本書共分十六章，第一章為電算機簡介，第二至第十二章為FORTRAN 原理及說明並以程式範例配合解說，每章之後並附有習題，以供讀者演練，第十三章為資料排序，十四章為「模擬」之研究，第十五章為三個案研究，最後一章則請讀者就三個案自行練習，以收舉一反三之效。

除簡易之單一敘述式問題外，本書各章習題大多可迎合商科學生興趣，因習題以商業導向之問題居多，並再次鼓勵學生以流程圖方式解題，書之最後並有五

HxA379/02

個附錄提供相關資訊以利學生快速查閱及參考。

本書之完成，承蒙本人同事及許多友人鼎力相助甚多，實難個別一一致謝，謹在此略表衷心感謝之忱，另特別感謝本人助理 Robert Nault 及提供許多建議而對本書之撰寫有相當貢獻的人士，包括 Dr. Richard Peddicord 等人，最後感謝我的外子給予的支持與鼓勵以及出版商樂於付印，本書方能順利出版。

譯序

本書係以 Nesa Labbe Wu 原著之 “ANSI FORTRAN IV and FORTRAN 77 Programming with Business Applications” 一書譯述而成，原著曾用於大專院校商學與電算科系有關程式設計課程，頗獲好評。

綜觀本書特點有三：

1. 強調邏輯觀念與流程方式解題。
2. 內容簡明，取材實用，程式範例具參考價值。
3. 注重程式設計技巧。

因此，本書甚適合供作大專院校初學程式設計課程之教材，亦可供電腦實務者或有志從事電腦工作者參考之用。

中文譯本已將原著若干未盡完善或重覆之處，均予以補充或修正，另近年來個人電腦（PC）使用甚為普遍，為期使本書亦可適用於PC機器上特另補加附錄一節，介紹有關PC FORTRAN並將原書範例擇要在IBM PC執行後，將程式與結果列印以供讀者參閱。

譯述工作期望能至「信、達、雅」之境界，誠非易事，而欲將科技原著，譯成詞意並茂本國語文，尤感不易，譯者才疏學淺難務紛忙僅利用公餘之暇疾筆振書倉促而成，疏漏錯漏在所難免，尚祈諸位先進不吝指正，不勝感激。

最後，本書之完成，承蒙大同公司資訊處楊翠華及目前在美進修之楊嘉蕙二位協助甚多，另外蘇士媛、許淑芝亦於百忙中，協助校稿與整理，在此深致謝忱。此外，松崗公司吳守信經理協助出版，以及內子許瑩輝的鼓勵照顧並提供溫馨舒適寫作環境，均表由衷的謝意。

目 錄

原著序

譯 序

第一章 電子計算機概論..... 1

1 計算機簡史..... 1

2 程式語言..... 3

3 計算機分時..... 5

4 計算機如何工作？..... 8

習題一..... 15

第二章 FORTRAN 程式簡介..... 17

1 處理 FORTRAN W 工作..... 18

2 原始敘述式..... 22

3 FORTRAN 程式編寫..... 22

習題二..... 26

第三章 流程圖..... 31

1 程式設計流程圖所用之符號圖形..... 33

2 範例(1)一位學生費之計算..... 39

3 範例(2)汽車貸款每月應付款項..... 43

2 FORTRAN 程式設計與商業上應用

4. 範例(3)房屋所有人財產稅·····	45
5. 邏輯流程之核驗·····	45
習題三·····	50

第四章 算術設定敘述式····· 57

1. 變數名稱·····	58
2. 算術運算式·····	59
3. 算術運算規則·····	61
4. 算術運算設定敘述之範例·····	67
5. FORTRAN 77 字元常數及設定式·····	69
習題四·····	72

第五章 輸入/輸出敘述式····· 77

1. 輸入/輸出敘述·····	77
2. 格式·····	78
3. 讀及其格式形態·····	79
4. 寫及其格式形態·····	85
5. FORMAT 之規則·····	97
6. FORTRAN 77 附加輸入/輸出·····	99
7. FORMAT 規格摘要·····	101
習題五·····	102

第六章 初級商業程式····· 107

1. 中止敘述·····	108
2. 簡易商業程式範例·····	109
習題六·····	120

第七章 轉移與決策敘述····· 125

1. GO TO 敘述式·····	125
-------------------	-----

2. 算術式 IF 敘述.....	127
3. 計值式 GO TO敘述.....	130
4. 邏輯式 IF 敘述.....	133
5. FORTRAN 77 附加之控制敘述式.....	144
6. 較佳之程式設計風格 (續前).....	148
習題七.....	149
第八章 迴路	161
1. 迴路.....	161
2. 迴路規則.....	165
3. 範例.....	170
4. FORTRAN 77 附加之迴路.....	177
5. 較佳之程式設計風格 (續前).....	179
習題八.....	180
第九章 註標之使用	187
1. 單註標變數.....	187
2. 雙註標變數.....	205
3. FORTRAN 77 附加之註標.....	222
習題九.....	224
第十章 更高層次之商用程式	231
1. 借貸款每月應付款項之計算.....	232
2. 庫存貨品之更新.....	237
3. 信用貸款 (週轉金) 計劃.....	243
4. 儲蓄累積—單身人壽保險.....	250
習題十.....	260

第十一章 FORTRAN 副程式..... 267

- 1 內儲函數或庫存函數..... 268
- 2 算術敘述式函數..... 271
- 3 FORTRAN 函數副程式..... 273
- 4 FORTRAN 次常規劃程式..... 278
- 5 COMMON 敘述式及變數維陣..... 283
- 6 副程式之較佳設計風格..... 285
- 習題十一..... 288

第十二章 型態宣告和資料敘述..... 293

- 1 整數宣告..... 293
- 2 實數宣告..... 295
- 3 雙精度宣告..... 295
- 4 邏輯宣告..... 300
- 5 隱含 (IMPLICIT) 敘述式..... 302
- 6 有關使用宣告敘述式之規則..... 303
- 7 資料 (DATA) 敘述式..... 304
- 習題十二..... 307

第十三章 資料排序..... 309

- 1 第一個程式：選擇排序..... 309
- 2 第二個程式：氣泡式排序..... 313
- 3 其他的排序技巧..... 316
- 習題十三..... 317

第十四章 模 擬..... 319

- 1 亂數產生器..... 319

2. 醉漢過橋模擬.....	322
3. 模擬方式計算 π 值.....	326
4. 運輸成本分析之模擬.....	331
習題十四.....	339
第十五章 商業分析案例	341
1. DOW JONES 工業公司股票之異動平均值分析.....	341
2. 亂數產生器以適合度法測試.....	348
3. 網路分析：計劃評核術.....	355
第十六章 三個案例研究	365
1. 個案 # 1：薪資給付實例.....	365
2. 個案 # 2：資產折舊實例.....	368
3. 個案 # 3：廣告分析實例.....	372
附 錄	381
A：流程圖符號.....	381
B：FORTRAN 敘述式摘要.....	383
C：流程圖列印.....	387
D：IBMO 29 穿孔機介紹.....	389
E：檔案處理.....	395
F：PC FORTRAN 及其範例簡介.....	401

第一章 ·

電子計算機概論

(Introduction of Computers)

1-1 電子計算機的發展史 (A Brief History of Computers)

計算機的起始，通常而言，已知最古老的計算機器大約於三千五百年前，在英國 Stonehenge 城有個用大石頭排成的圓圈，被大多數的科學家認為是最早的計算機。它是配合太陽來預測一些天體現象。另外有關於較古老計算機的報導可在 1959 年 6 月出版的“ Scientific American ”雜誌上找到。那是一個有關於希臘時期的計算機器，在希臘 Antikythera 島外所發現的。估計大約有兩千年以上歷史。另外一個早期的計算機器便是中國人發明的算盤，這種簡單計算工具約在馬可波羅時代傳入歐洲。

第一部加法機器是由法國著名的數學家 B. Pascal 在 1642 年發明的，此種計算器在 1673 年時代經德國數學家 Leibnitz 改良，使得這部機器不僅能做加法運算，同時亦能做乘法運算。

Charles Babbage 常被尊稱為電子計算機之父，他是英國劍橋大學一位有名的數學教授，他在 1812 年時曾構思一部能操作指令的機器稱為「差分機」(Difference Engine)，這是一部體積巨大，重約二噸的龐然大物，但由於當時的技術並未成熟，故此部機器並未被完成。稍後 Babbage 發展出一部「解析器」(Analytical Engine)，可惜由於財務上的困難，功虧一簣，阻礙工作之完成，他想像其發明的機器能一次執行一個算術運算，亦構想從打孔卡片 (Key-punched Cards) 上讀取指令，儲存資料，並且能於必須二擇一時作一決定，此

解析器所提出之內儲程式 (Store program) 觀念，非常接近於現代的電子計算機的概念。

第一部電算機是「馬克 1 號」電算機 (Harvard Mark 1)，完成於 1944 年，係由 H. Aikens 在哈佛大學 (Harvard U.) 於 1937 年開始設計完成，能自動執行一連串的數學運算，雖然馬克 1 號的大小只有 Babbage 所構想機器之十分之一，卻仍然過於龐大，它的內部包含有加法器 (adding accumulators)，繼電器 (mechanical relays)，開關 (switches)，按鈕 (buttons)，電線插頭 (wire plugs) 和打孔紙帶 (punched tape)。其後不久，J.P. Eckert 和 J.W. Maughly 在賓州 Moore 工程學院完成了 ENIAC (1945) 和 EDVAC (1952) 兩套電子計算機。

用二進位數字系統來表示資料和指令是由在普林斯頓大學 (Princeton V.) 任教之 D. J. Von Newman 博士所提出 (其後亦在 Moore 學院任教)，並於 1952 年完成 IAS 電算機。從 EDVAC 計算機和 IAS 計算機起，計算機便從機械時代進入電子時代。

我們將計算機依其在硬體技術上的演進區分為四個階段 (或稱“四代” (generations))。第一代計算機 (1946 — 59) 使用真空管，速度較慢，第二代計算機 (1959 — 65) 使用電晶體，體積上小了甚多，並且執行速度亦較快。

第三代計算機 (1965 — 70) 其特性為使用積體電路並且具有分時 (time-sharing) 能力。第三代計算機的降臨亦促成了計算機工業上的一大改變。計算機之功能、速度越來越增進，惟價格卻下滑甚多。此種價格上滑落或許可解釋為何現有百分之六十五的計算機都是於六十年代所裝設的。第四代計算機 (1970 —) 是使用今天所謂的微電子技術。迷你型計算機 (Minicomputers) 和許多預期的發展如衛星傳輸 (Satellite transmission)，微電算機 (Microcomputers)，雷射 (Laser) 和 Fluidic 計算機，以及其他許多的發展都是這一時代的特色。

軟體的發展則直接反映在許多不同計算機系統上，所發展的一些程式，第一代計算機使用的是機器導向語言 (machine-oriented language)，但這種語言並不實用，這是由於各個計算機系統有其特有之機器語言，因而不能互通使用

。而第二代計算機則使用所謂的組合語言（Symbolic assembly language），雖然這種組合語言較容易撰寫，但卻由於在衆多的使用者中缺乏其一致性，因而阻礙了它的發展，最後編譯程式（compilers）和一些高階語言（high-level language）便被發展出來，高階語言的優點在於其獨立性與運用性，亦即不同廠牌計算機也能互通使用，並且較前述的任何語言更容易撰寫。FORTRAN 程式語言便是最早的高階語言之一，由 IBM 發展出來並在 1957 年公開上市，期間並經過了多次修改，最廣泛使用的版本為 FORTRAN IV，本書將討論 FORTRAN IV 及最新版即 FORTRAN 77。

電子計算機有兩種不同主要類型：類比計算機（analog computer）和數位計算機（digital computer）。類比計算機主要涉及到連續物理量（continuous quantities），數位計算機則只涉及至分離物理量（discrete quantities），在早期約 1600 年代中，第一套廣泛使用的類比電子計算機為計算尺（slide rule）。1872 年 Lord Kelvin 建造出一部大型（large-scale）的類比計算機，用來預測英國港灣海水的潮汐變化。類比計算機不如數位計算機般地廣泛使用，並且也不能使用於像 FORTRAN 等的程式語言。

1-2 程式語言（Programming Languages）

在程式語言中主要可區分為二種不同之類型，一種為機器導向語言（Machine Oriented Language），另一則為問題導向語言（Problem Oriented Language）。不論何種類型的語言，皆為讓使用者能與計算機溝通之工具。

在所有的語言中，最基本而直接的語言便是機器語言（Machine Language），在機器語言中，命令是由一連串的「0」與「1」所組成。在語言編譯器（language translators）還沒發展出來以前，所有的程式都是用機器語言或「目的碼」（object code）來撰寫。由於機器語言牽涉到計算機硬體（hardware）其本身一些特殊功能，並且須耗費甚多時間學習，不適合於一般使用者及程式人員使用，因此許多的語言編譯器便因應此等需求而漸次發展出來，以解決此一問題，組合語言（Assembly Language）即為其中一例。

在組合語言裡使用一些符號文字 (symbolic words) 稱作記憶碼 (mnemonics) 來直接和系統溝通。從機器觀點而言，使用組合語言執行程式是最經濟有效率的，但是從程式設計員觀點而言，卻並非如此，因為一般應用程式設計人員所需要的是最能符合他們的應用語言，於是問題導向語言便陸續的被發展出來。

下面吾人將詳細討論一些問題導向語言如：FORTRAN，COBOL，RPG，PL/I，BASIC，ALGOL，GPSS 和 SIMSCRIPT。

福傳 (FORTRAN) 程式語言是由 IBM 所發展並在 1957 年發表。FORTRAN 是由 FORMula TRANslator 演譯而來，最初設計之目的是用來解決數學運算的問題。目前最被廣泛使用的版本是 FORTRAN IV，由美國國家標準學會 ANSI (American National Standards Institute) 在 1966 年發表公布其標準，1978 年 10 月。在 ACM (Association for Computing Machinery) 雜誌上發表了關於將控制單元與算術邏輯單元予以分開成各個獨立的實體。因此這五個基本單元 FORTRAN 語言最新報告，提出了一個新標準版本 “ American National Standard Programming Language FORTRAN，X3.9-1978 ”，一般稱作 FORTRAN 77，這個新標準版本加進了許多以前從未有的功能及特色，並使得程式較容易轉換，注意 FORTRAN 77 所具有的特點並沒有與 1966 年發表的 FORTRAN 有所衝突。本書將討論 FORTRAN IV 和 FORTRAN 77 所具有之新功能。過去幾年由於 FORTRAN 經過了多次的修改，使得 FORTRAN 不能處理數學和公式上的一些問題，並且亦具備了處理字串文句之能力，因而 FORTRAN 被廣泛的使用在數學、教育、商業、工業的應用上。

1960 年一種新的語言 COBOL (Common Business Oriented Language) 被引介問世，COBOL 指令是由類似英文之句子 (sentence) 及段落 (paragraphs) 所組成。當初設計之目的是用來作商業上的資料處理，因而成爲一種被廣泛應用之商用語言。COBOL 需要使用較多指令，因而較 FORTRAN 難學。

RPG (Report Program Generator) 與其較進步之版本即 RPG II，係於 1969 年發展出來，是特別爲了商用報告印表所設計，程式設計員只須指定一些

輸入與輸出的資料及所要之計算，RPG II 編譯器便能自動地產生結果。此種語言常在小型的系統上使用，在中型或大型的系統上較少使用。

1966 年 IBM 發展出一套新的程式語言，兼具有 COBOL 在商用處理的能力與 FORTRAN 在數字處理的能力。這套語言稱作 PL/I (Programming Language I)，並且假設所有計算機製造商皆能接受此一語言，但時至今日卻只有極少數的計算機製造廠商成功地發展出 PL/I 編譯器 (compiler)，因此 PL/I 便不像 FORTRAN 或 COBOL 那麼頻繁的被使用在一般的應用上。

1964 年，BASIC (Beginner's All-purpose Symbolic Instruction Code) 在 Dartmouth 學院由 John G. Kemeny 和 Thomas E. Kurtz. 之領導下發展出來。這套語言在第四代計算機中的微計算機及分時系統上廣受歡迎。它是為一般的初學者所設計的，它像 FORTRAN 一樣亦是一種代數式 (algebraic) 語言，但是比 FORTRAN 多了許多限制。現今它已經變成初學者或微計算機上最主要的語言。

FORTRAN 是由美國所發展出來的，而 ALGOL (ALgebraic Oriented Language) 則是由歐洲所發展，雖然 ALGOL 與 FORTRAN 非常類似，但卻不像 FORTRAN 般廣泛地被接受。

另外二種值得一提的問題導向語言是 GPSS 和 SIMSCRIPT。SIMSCRIPT 主要是應用於大型而複雜系統的模擬 (simulation) 上，並且減少其在設計程式規則及測試模式上所花費的時間，它是一種無特定格式而類似英文 (English-like) 之語言，由 RAND 公司發展出來。GPSS (General Purpose Simulation System) 程式語言是在 1960 年代的早期發展，它是特別為了系統上的模擬所設計，較 SIMSCRIPT 容易學習容易使用。

新的計算機語言將繼續被發展，而既有的程式語言亦將逐漸地改進，以符合未來更廣泛之應用，並使程式之設計變為更簡易。

1-3 計算機分時 (Computer Time - sharing)

分時系統背景簡述：

所謂「分時」便是二個或多個使用者能同時至同一計算機系統上存取資料並得到結果。雖然實際上機器是將時間切割成幾個小片段分配給使用者，但使用者卻仍覺得機器在自己完全的控制下運作。縱然使用者離中央計算機（central computer）有一段距離，仍可透過終端機（terminal）、電話線（telephone line）與中央計算機取得聯絡。第一套實際運作的計算機分時系統是由麻省理工學院在十年前引入。稱作「相容式分時系統」（Compatible Time-Sharing System, (CTSS)），於1961年在IBM 709型計算機上發展。其他的分時系統在往後幾年亦陸續被許多終端使用者（End-User）、計算機製造商及一些商用的分時系統銷售者所裝設。其他一些早期的系統有：

CTSS：由Bolt, Beranek, Newman等在PDP-1型計算機上發展。

JOSS：（Johnniac Open Shop System）由RAND公司於1963年發展。

CAL：（Conversational Algebraic Language）由加州大學柏克萊分校於1963年發展。

MAC：（Multiple Access Computer）最初的大型分時系統，於1963年發展出，共有160部終端機裝設在MIT內，及其他高級研究學術單位。

BASIC：由Dartmouth學院在1964年至1966年間發展的。

許多分時系統是由主要計算機製造廠商所發展出來，如：Burroughs, Control Data Corporation (CDC), Digital Equipment Corporation, General Electric (GE), Hewlett-Packard (HP), IBM, RCA, Scientific Data Systems, 和UNIVAC等。

在1970年代，超過十億美元花費在分時系統研究上，因此分時系統已是計算機工業上發展最迅速的一部分。使用分時系統者包括有各學院、大學、科學家、工程師、製造商、經理人員、銀行人員，及其他許多各行業的使用者。

終端機，如一般的電傳打字機（teletype）、顯示器（display）、繪圖器（plotter），常可在許多辦公室或一些私人家庭裡發現。它們在商業應用上有許多用途如一投資分析、運輸、回收效率、計劃評核、財政分析、合併分析、

庫存管理、採購、預算、資料處理，及其他許多用途。分時系統亦被許多單位廣為接受。政府機構、公共衛生服務組織，及其他一些社會團體等。

分時系統對小型的企業特別具有吸引力，因為初期的投資是最主要的考慮因素，但是不管如何最主要的使用者仍舊是大型的公司，因為回返時間（turnaround time）【註】：回返時間是指傳送工作給計算中心，再送回結果所費的時間，才是最重要的因素。

分時性語言（Time-Sharing Languages）

一些交談式的分時程式語言包括有：

BASIC：一種較簡易的語言，大多數的計算機製造商都有提供。

CAL：由加州大學柏克萊分校所發展，較不普遍，可以在 Scientific Data 940 上執行。

LISP：一種串列處理（LIST-Processing）程式語言，在許多系統上都能使能。

在許多版的 FORTRAN 程式語言被用來發展分時系統，如：QUIKTRAN，和 XTRAN，和 CDC FORTRAN EXTENDED。

遠程工作輸入（Remote Job Entry VS.）及交談式處理（Interactive Processing）：在一個遠程工作輸入環境裡，程式與資料皆是製成卡片並委託一些中心收集再送入整批處理系統（batch processing system）中處理，所有的使用者所看到的僅是讀卡機和列表機而已。

在交談式處理系統（Interactive processing system）環境下，程式和資料都是藉著線上（on-line）終端機連線作業直接進入中央計算機。

本書主要以製卡（整批）處理系統，教師亦可提供一些額外之補充，關於交談式處理系統，對於 FORTRAN 來說亦復如此，差別只是在於學生如何使用這套電子計算機系統。