

大學用書

FORTRAN 程式語言

劉振漢著

三民書局印行

FORTRAN程式語言

劉 振 漢 著

學歷：美國加州柏克萊大學計算機博士
現職：國立交通大學計算機研究所教授



三 民 書 局 印 行

中華民國七十年二月初版

FORTRAN 程式語言

基本定價叁元壹角壹分

行政院新聞局登記證 版業字第 0020 號

版權所有
翻印必究

著者 劉振漢
發行人 劉振強

出版者 三民書局股份有限公司

印刷所 三民書局股份有限公司
臺北市重慶南路一段六十一號
郵政劃撥九九九八號

序

計算機的大量應用已有二十年的歷史，在這二十年中雖然不斷有新的計算機語言的出現，一直都沒有取代 FORTRAN 和 COBOL 這兩種語言。COBOL 的主要用途在商業和公務方面，像公司會計、保險業務、稅務處理等等；FORTRAN 則最初用在科學計算方面，而在後來也被用來寫操作系統 (operating system) 和編譯器 (compiler)，而商業上應用的程式也漸漸有人用 FORTRAN 來寫，在使用範圍上，FORTRAN 比 COBOL 又更為廣泛。

FORTRAN 是第一個問世的高階計算機語言 (high level computer language)，由 IBM 公司在 1957 年推出，立即成為無明文規定的標準語言，所有的計算機都備有 FORTRAN 的編譯器，而學習 FORTRAN 和學習計算機幾乎是同義的說法。到 1966 年美國國家標準局才公佈制定 FORTRAN 的標準規格，比起 IBM 當年所推出的 FORTRAN 進步之多，已不可相比。然而隨着計算機技術的進步，各計算機廠商不斷地將新的能力加入他們的 FORTRAN 中，到 1975 年左右，FORTRAN 程式由一個計算機轉換到另一個計算機已變為很困難的工作，很明顯地有必要將 FORTRAN 制定新的標準規格，容納計算機新的能力，滿足使用者新的需求。在 1976 年美國標準局出版了一本新的 FORTRAN 規格書稿，接受各方建議，1977 年公佈修訂稿，1978 年正式宣佈為新的規格，世稱 FORTRAN 77。

今日各計算機所提供的 FORTRAN 都是 FORTRAN 77 規格制定的前後期間製作的，和 FORTRAN 77 的差異均很微少，未來的 FORTRAN 當然更會以 FORTRAN 77 為主要根據。本書所述即以 FO-

2 FORTRAN 程式語言

RTRAN 77 為基準，與目前各計算機上 FORTRAN 相比，幾乎沒什麼差別，換句話說，本書可以做為使用任何計算機上的 FORTRAN 的參考手冊，不會有太多不足的地方。然而，本書最主要的目的是做為一個教科書，介紹 FORTRAN 語言給初學計算機的讀者，在舉例解說、製作習題方面特別費心，絕大部份的例題和習題都是新作。

已識 FORTRAN 而欲查明 FORTRAN 77 中新添項目者，可以藉書後索引查尋。書後附錄 B 列舉了 FORTRAN 77 與舊的 FORTRAN 不同之處，可做一比較。

劉 振 漢

70年1月 於新竹

FORTRAN 程式語言 目次

序

第一章 簡 介

1-1 計算機內資料的儲存.....	1
1-2-1 整數(integer)表示方法.....	3
1-2-2 實數(real number)表示法.....	5
1-2-3 符號(character)表示法.....	7
1-2-4 指令(instruction)的儲存.....	8
1-3 FORTRAN 程式的形式.....	12
1-4 程式的執行.....	13

第二章 算術運算式和存入指述

2-1 算術常數(arithmetic constant)	17
2-1-1 整數(integer number).....	17
2-1-2 實數(real number)	18
2-1-3 倍準數(double precision number).....	19
2-1-4 複數(complex number)	20
2-2 變數(variable)	21
2-3 變數的名稱(variable name).....	22

2 FORTRAN 程式語言

2-4 變數資料類別.....	23
2-5 IMPLICIT 宣告指述 (IMPLICIT statement)	23
2-6 個別變數類別宣告語句 (data type statement).....	24
2-7 變數類別綜合規則.....	25
2-8 存入指述 (assignment statement)	26
2-9 算術運算式 (arithmetic expression).....	26
2-10 混用算術運算式	31
2-11 轉換資料類別的函數	32
2-12 乘方的處理	34
2-13 整數的除法	35

第三章 變數陣和計次反複

3-1-1 資料的輸出入 (data input and output).....	41
3-1-2 數列引導式輸出入 (list-directed I/O).....	41
3-2-1 變數陣 (array)	45
3-2-2 變數陣的儲存.....	47
3-3-1 計次反複 (DO loop)	48
3-3-2 計次反複的通式.....	52
3-3-3 多層計次反複 (nested DO loop).....	54
3-3-4 指數的設定.....	56
3-3-5 暗喻計次反複 (implied DO loop)	57
3-3-6 矩陣的乘法 (matrix multiplication).....	60

第四章 邏輯運算式和條件控制

4-1 邏輯常數和邏輯變數 (logic constant and logic variable)	67
4-2 關係運算式 (relational expression)	68
4-3 邏輯運算式 (logical expression)	70
4-4 邏輯式假設指述 (logical IF statement)	75
4-5 無條件 GO TO 指述 (unconditional GO TO statement)	77
4-6 結構式 IF 指述 (structured IF statement)	81
4-7 多層結構	90
4-8 算術式 IF 指述 (arithmetic IF statement)	92
4-9 計值 GO TO 指述 (computed GO TO statement)	94

第五章 符號資料的處理

5-1 符號常數 (character constant)	105
5-2 符號變數 (character variable)	107
5-3 符號變數的存入指述	108
5-3-1 符號變數的子串 (substring)	108
5-3-2 符號運算式 (character expression)	109
5-3-3 符號存入指述 (character assignment statement)	110
5-4 符號關係式	112
5-5 符號資料的輸出入	114
5-6 處理符號串的內在函數	117

第六章 資料輸出入

6-1 資料輸出入概念.....	127
6-2 完整的輸出入指述.....	128
6-3 FMT 參數的種類和出現形式	134
6-4 FMT 參數的意義	137
6-4-1 格式碼的分解說明.....	140
6-4-2 X 格式碼.....	142
6-4-3 整數的格式碼.....	143
6-4-4 邏輯變數的格式碼.....	145
6-4-5 符號變數的格式碼.....	146
6-4-6 符號常數的格式碼.....	146
6-4-7 T 格式碼.....	147
6-4-8 實數的格式碼.....	148
6-4-9 倍準數的輸出入.....	152
6-4-10 複數的輸出入	153
6-4-11 結束記錄的格式碼	153
6-5 反複格式碼的簡寫.....	155
6-6 變數序列、格式碼和記錄的對應關係.....	156
6-6-1 FMT 參數的反復使用 (format rescan).....	156
6-6-2 多餘格式碼的作用.....	159
6-7 程式範例.....	162

第七章 副 程 式

7-1 副程式 (subprogram) 簡介.....	169
7-2 內函數.....	172
7-3 程式元.....	175
7-4 函數副程式.....	177
7-5 函作副程式.....	184
7-6 實據數和虛據數的對應關係.....	189
7-6-1 實據數為一簡單變數 (非變數陣)	189
7-6-2 實據數為一常數.....	190
7-6-3 實據數為一運算式.....	191
7-6-4 實據數為一變數陣名稱或變數陣成員.....	192
7-6-5 實據數為一符號串.....	198
7-6-6 實據數為一副程式名稱.....	199
7-6-7 實據數為一標號.....	202
7-7 副程式的不同進入點 (multiple entry points for subprograms)	207
7-8 單指述函數 (statement function).....	210

第八章 宣告指述

8-1 EQUIVALENCE 指述	219
8-2 DATA 指述.....	226
8-3 PARAMETER 指述	229
8-4 COMMON 指述.....	230

6 FORTRAN 程式語言

8-4-1 共用區與據數的關係.....	234
8-4-2 標名共用區 (named common block).....	238
8-4-3 COMMON 指述和 EQUIVALENCE 指述的關係	240
8-5 BLOCK DATA 指述和 BLOCK DATA 副程式	242
8-6 程式元中各類指述的順序.....	244

第九章 高級輸出入

9-1 輸出入參數.....	251
9-2 OPEN 指述	252
9-3 CLOSE 指述	256
9-4 檔案類別.....	257
9-4-1 formatted (機外形式) 和 unformatted (機內形式) 檔案.....	257
9-4-2 循序檔案 (sequential access file) 和直入檔案 (direct access file)	260
9-5 廣義的檔案.....	265
附錄 A.....	268
附錄 B.....	269
索 引.....	273

73.87221
42

第一章 簡 介

1-1 計算機內資料的儲存

計算機內，最小的記憶單位是爻元(bit)。一個爻元是一個微細的電子結構，可以穩定地停留於兩種狀態。一個相近的例子是各種電器的開關，這些開關都可以停在兩種狀態，“開”的狀態和“關”的狀態。計算機內爻元的兩種狀態，都以“0”狀態和“1”狀態來稱之。爻元的狀態可以很容易用電子信號加以改變或加以測出。

計算機的設計都將固定數目的相鄰爻元集成一稍大的記憶單位，叫做字(word)。每一字的爻元個數視計算機機型而不同，常見的有 8, 12, 16, 24, 32, 36, 60 等。目前最常見的是每字含 16 爻元和 32 爻元。圖(1)顯示一個含 16 爻元的字。

2 FORTRAN 程式語言

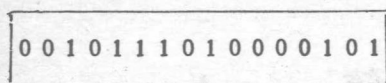


圖 (1) 一個含16爻元的字

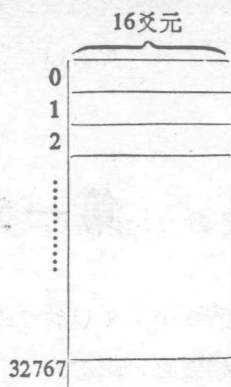


圖 (2) 一個含32768字(32KW)的記憶

一個字內各爻元所處狀態，就是這個字的所“存”（或說所“記憶”）的“資料”（或說是“值”）。圖(1)的字存的值為 0010111010000101。一個字內所存的值可以用電子方法將其測出來，這叫做“讀”這個字的資料，讀後字的值不變。也可以將外面準備好的資料存入一個字，舊的資料就不見了，這叫做將資料“寫”入這個字。讀和寫的作用，和錄音機的“放”和“錄”的作用相似。

一個計算機內含的字的數目通常在幾萬個字到上百萬個字。每一個字有其位址。圖(2)顯示一個含 32768 字的記憶 ($32768 = 2^{15} = 2^5 \times 2^{10} = 32 \times 2^{10} = 32K, K = 2^{10} = 1024$)。

計算機所處理的資料主要有三種：數字、文字、指令。一個字用來存一個完整的資料。例如一個每字32爻元的機器裏，一個字通常用來存一個整數，一個實數，四個符號，或一個指令等等；一個倍準數則需要兩個字才能存得下。每字含多少爻元，可以存多少資料，因計算機種類而不同。目前的計算機有每字16爻元，24爻元，32爻元，36爻元，60爻元等等。

在下面的幾個小節裏，我們簡單地以16爻元機器為例，說明資料的

表示方法。不同計算機各自有其一套表示方法，因此以下所述，不過是一概念而已。在“組合語言”和“計算機結構”的書本中有較詳盡的介紹。

1-2-1 整數 (integer) 表示方法

整數的存法，可以最左的爻元表正負號，0 爲正，1 爲負，剩下的 15 爻元表絕對值。絕對值是 2 進位 (binary)，由右至左每一位的值分別爲 $2^0, 2^1, 2^2, 2^3, \dots, 2^{14}$ 。

一個字內含 $d_{15}d_{14}d_{13}\dots d_0$ ，其中每個 d_i 爲 0 或 1，則其代表的整數值爲

$$(-1)^{d_{15}} \times (d_{14} \times 2^{14} + d_{13} \times 2^{13} + \dots + d_0 \times 2^0)$$

註： $(-1)^1 = -1$ ， $(-1)^0 = +1$ ， $2^0 = 1$

例如一個字所含值爲 100010010111001，如將其以整數看待，則其值爲

$$\begin{aligned} & (-1)^1 \times (1 \times 2^{11} + 1 \times 2^8 + 1 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^0) \\ &= - (2048 + 256 + 64 + 32 + 16 + 1) \\ &= -2417 \end{aligned}$$

反過得，要得出一個整數在計算機內的存法，可以將此數連續用 2 除，所得各餘數即是所求。

例如絕對值 395 一數，連續被 2 除，結果如下：（中間爲商，右邊爲餘數，左邊爲除數。）

4 FORTRAN 程式語言

2	395	1
	197	1
	98	0
	49	1
	24	0
	12	0
	6	0
	3	1
	1	

因此 395 的 2 進位表示法爲

1 1 0 0 0 1 0 1 1

存入計算機一個字時，將其左邊補 0，加上正負號即可，因此 395 和 -395 存入計算機中的一個字，分別爲

0 0 0 0 0 0 0 1 1 0 0 0 1 0 1 1

1 0 0 0 0 0 0 1 1 0 0 0 1 0 1 1

各種計算機大多以一個字（也有用半個字或兩個字的）來存一個整數。一個整數，不管其數值大小，都佔一個字。其能處理的整數的大小，即由其每字的爻元數目來定。一個存於 n 爻元的整數之數值的極限

在正負 $(2^{15}-1)$ 之間。例如存於 16 爻元的字能存整數在 $-(2^{15}-1)$ 到 $(2^{15}-1)$ 即 -32767 到 32767 之間。

因此使用計算機而必須處理絕對值很大的數字時，就得注意到所使用計算機的數值極限。

1-2-2 實數 (real number) 表示法

實數即是含有整數部份和分數部份的數，例如 645.78 即是。實數在計算機中存法和整數很不相同，通常是以一個或兩個字來存。

一個存在計算機內的實數分三部份：符號 (sign)，冪 (exponent)，和尾數 (mantissa)。一個 16 爻元計算機的實數可能是如下設計：

S	E	M
1	2-7	8-16

其所代表的值為

$$(-1)^S (M \times 2^E)$$

其中 S 代正負號，0 為正，1 為負。E 是冪數，其值在 -31 在 $+31$ 之間。而 M 是尾數，是小數點在“最左邊”的 2 進位數。

例如下面的字，如以實數看待，則得

1 0 0 0 1 0 1 1 0 1 1 0 0 0 1 0

$$S = 1 \text{ (負數)}$$

$$E = 101 = 5$$

$$M = .101100010$$

6 FORTRAN 程式語言

$$\text{實數值} = -(.101100010 \times 2^5)$$

$$= -10110.0010$$

$$= -(2^4 + 2^2 + 2^1 + 2^{-3})$$

$$= -(16 + 4 + 2 + .125)$$

$$= -22.125$$

反過來，如果已知實數 25.37，要求其在計算機內的存法，則可按以下的步驟求得：

(1) 求此數的 2 進法表示方法

(整數部份)

2	25	1
	12	0
	6	0
	3	1
	1	

整數部份為 11001

(分數部份)

	37	× 2
0	74	
1	48	
0	96	
1	92	
1	84	
1	68	