

BASIC

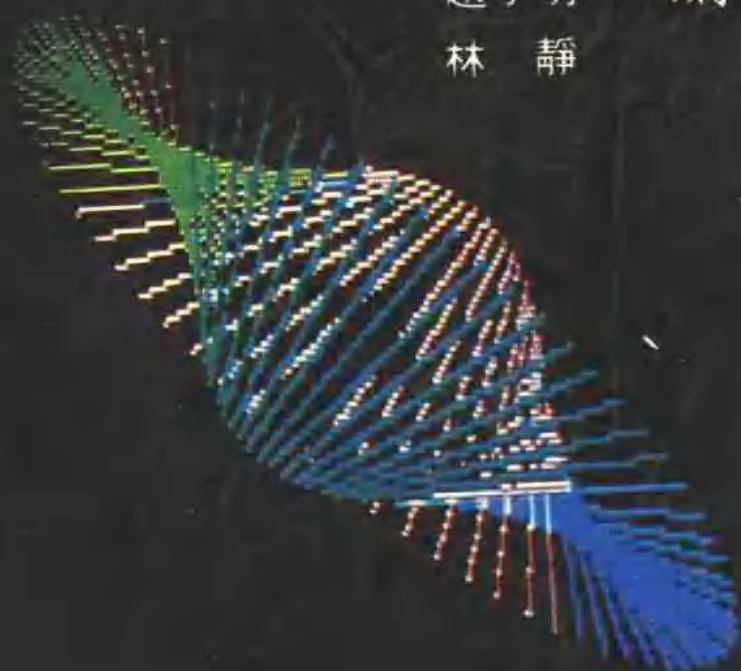
程式設計

謝耀華

趙子明

林 靜

編 著



中央圖書出版社出版

BASIC

程式設計

編 著

趙 子 明

國立台北商業專科學校講師

謝 耀 華

私立銘傳女子商業專科學校講師

林 靜

東方電腦中心負責人

中央圖書出版社出版

行政院新聞局出版事業登記證
局版台業字第〇九二〇號

BASIC 程式設計

完必印翻・有所權版

整元拾伍佰壹幣券新價實

編著者：趙子耀

謝林

出版者：中央圖書出版社

台北市重慶南路一段一四一號

發行人：林在

中央圖書供應社
台北市重慶南路一段一四一號

電話：

三三一五七二六
三七一八九三

郵政劃撥帳戶：九一四號

印刷所：利康印刷有限公司

三重市民生東街廿六之一號

中華民國七十二年二月初版

中華民國七十二年八月修訂版

編號：2851

序

H. J. A. S. T. C. 註冊會計師(公)

• 244 • 附录二

目 錄

序

第一章 概 述

第一節 計算機與程式語言	1
第二節 計算機系統	2
第三節 數系，位元，字元	3
第四節 計算機運算程序	8
第五節 程式規劃的幾個關鍵字	9

第二章 第一個步驟

第一節 儲存資訊	5
第二節 計 算	11
□第三節 進一步闡述 LET 敘述	17
第四節 標 題	19
第五節 註 解	24

第三章 程式的迴路

第一節 計數器	41
第二節 程式迴路的應用	43
第三節 迴路內累計總數	50
第四節 表 格	58
第五節 重覆的使用資料	65
第六節 不同長度的迴路	72
□第七節 巢狀迴路	78

VI 目 錄

第四章 變數，數值與輸出

第一節 變數與數值	93
第二節 輸出設計	98
第三節 定 位	101
第四節 圖 形	105
第五節 統計圖	114

第五章 條 件

第一節 IF 與 GO TO 敘述	125
第二節 迴路中的條件式	131
第三節 有條件的計算	138
第四節 超過一個條件式的程式	144
第五節 用資料結束的符號來處理資料	152
第六節 使用條件來終止迴路	158
□第七節 找出最大的數值	162
第八節 取整數值	168

第六章 交談式 (Interactive) 程式

第一節 INPUT (輸入) 指述	181
第二節 交談式程式	189

第七章 串列 (List)

第一節 因次	193
第二節 進一步使用串列來解題	207
第三節 計算頻率以繪製圖表	220

第八章 資料查詢

第一節 選擇某些資料	231
第二節 多條件的資料查詢	238

第三節 資料的分類·····	244
第九章 函 數	
第一節 一般用途函數·····	251
第二節 三角函數·····	257
第三節 定義函數·····	263
第四節 定義函數之應用·····	264
第十章 BASIC 摘要	
第十一章 程式專題	
專題 1 登月小艇著地模擬·····	277
專題 2 帳目管理·····	278
專題 3 排列數字·····	279
專題 4 追蹤飛機·····	281
專題 5 倉儲記錄·····	282
專題 6 瓦斯帳單·····	282
專題 7 築路計劃·····	283
專題 8 雜誌訂戶資料處理·····	284
專題 9 銀行帳目·····	285
專題 10 工業用地選擇·····	285
專題 11 學生選課資料·····	287
專題 12 排隊購票模擬·····	288
專題 13 人口成長·····	289
專題 14 太陽系·····	290
專題 15 薪資細目·····	291
專題 16 圖書資料·····	291
專題 17 十字路口交通模擬·····	292
專題 18 所得稅·····	294
專題 19 中央暖氣系統·····	295

VIII 目 錄

專題20 電腦藝術.....	297
----------------	-----

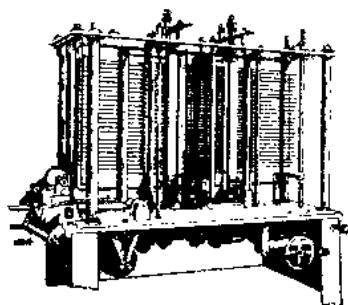
第一章 概 述

第一節 計算機與程式語言

電子計算機(Electronic computer)是現代科學技術最大成就之一。它能接受外來資訊(Information),記憶儲存,或與原來儲存資料(Data)共同處理,及對外輸出(Output)資訊等。這些功能在在都與人腦的作用相似,所以俗稱之為電腦。它的發展雖只有短短幾十年*,

* 計算器的歷史很早,可遠溯至希臘時代。我國的算盤也是很早的計算器。不過都沒有和現代的計算器發生太大的直接關係。

最早的計算機應該算英國數學家查理·巴貝(Charles Babbage, 1792-1871)於1833年所設計的分析機(Analytical engine)。由於當時工業條件不夠,巴氏並沒有完成這部機器。但現代計算機所應用的很多原理仍是以巴氏的設計作基礎。第一部數位計算機(Digital computer)是在1944年由美國哈佛大學教授霍華·艾肯(Howard Aiken)完成。是利用機械和電力控制。而利用真空管控制的數位計算機,却是在1946年由美國賓州大學依寇特(J. P. Eckert)及Mauchly與IBM(萬國商業機器公司, International Business Machines Co.)公司一批工程師們合作設計出來,稱為ENIAC(電子數字積分器及計算器, Electronic Numerical Integrator and Computer)。但真正作為商品推出於市場的是在1951年, J. P. Eckert 及 Mauchly 兩氏設計的UNIVAC(通用自動計算機, Universal Automatic Computer)。此後,由於電子科學與電腦技術互相影響,使計算機不斷地改進。在製造上,從使用真空管而電晶體而積體電路(Integrated circuits, 縮寫為IC)。使其體積從需要一個大房間才能安裝容納,減小到可放在辦公桌上或手提攜帶。而計算速度與記憶容量也遠非第一代電腦所能比;在程式語言方面也有新的發展,新的語言更容易學習和應用更便捷,也更符合使用者的各種要求。



分析機

却對人類生活有空前巨大的影響。在今天，無論政治、軍事、經濟、教育、或農、礦、工商各行各業莫不廣泛採用，而且，更介入各個家庭，直接與我們生活發生聯繫，甚至有人提出“不學電腦，等於電腦文盲”說法。

電腦在執行演算時，比人腦快百千萬倍以上，並可不停地工作。其「記憶力」（儲存資料）之強更遠非人腦所能及。這是它被廣泛接受的主要原因。不過，它雖然具備了很多人所不及的功能，但它只是供人類使用的工具；它不具備主動思考力，只能在人的操作和指令下，迅速執行和完成任務。

要計算機執行任務，就必須與計算機通訊，也就是與電腦交談。使它瞭解我們要它作些甚麼？或它答覆我們。為了達到這一要求，必須使用電腦能懂得的語言。目前，學術界發展出來用於電腦的語言有很多種，如：FORTRAN（福傳），COBOL（可博），ALGOL（奧高），BASIC（基礎程式語言），PASCAL（巴斯噶語言），Ada（艾達語言）Forth（第四代語言），C Language（C 語言）等。這些語言各有優點及其適用的領域。本書所介紹的是基礎程式語言。

BASIC 正確的譯名應該是“初學者多用途符號指令碼”。這是取 **Beginner's All-purpose Symbolic Instruction Code** 每一字的首字母集成的集義字*。它是由美國 Dartmouth 學院所發展成功的一種代數程式規劃碼。是一種會話型程式規劃語言（Programming language），具有人、機對話功能。它使用像英文的陳述及數學的記法。容易學習，容易精通。

第二節 計算機系統

計算機系統可概分為硬體（Hardware）和軟體（Software）。硬體指在計算機系統裏的金屬的或“硬”的組件。也就是所有自動資料處

*BASIC 也是 **Beginner's Algebraic-Symbol Interpreter Compiler**（初學者代數符號解釋程式），或 **Battle Area Surveillance and Integrated Communication**（戰區監督與集體通信）的縮寫。

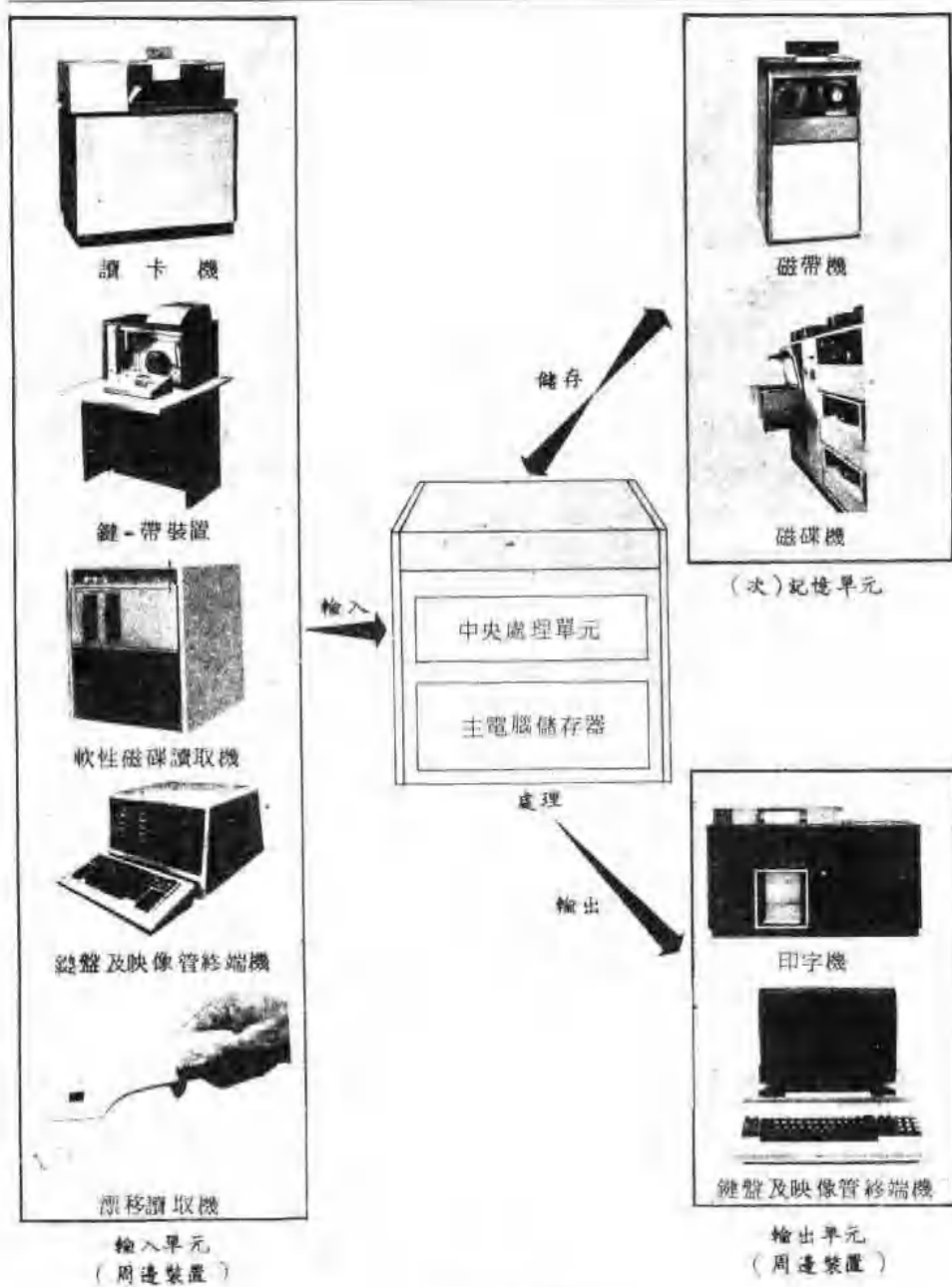
理設備，包括本體機件，磁、電或電子裝置（包括輸入、輸出裝置，運算與控制線路，記憶器等）。從它們的功能又可概分為三大部份：

1. 中央處理單元 (Central processing unit, 簡稱為CPU.) 是計算機的主要部份，等於人的頭腦，是控制與運算機構。它能遵照指令 (Instruction) 執行計算及控制輸入/輸出工作。
2. 記憶單元 (Memory unit) 等於人腦的記憶部份。記憶 (儲存) 資料及指令。
3. 周邊裝置 (Peripheral devices) 等於人的眼、耳、口。主要用來與外界交通。它們與主機連線結合操作，擔任輸入、輸出或其他工作。但不是計算機的零件。終端機 (Terminal) 包括鍵盤 (Keyboard) 及印字機 (Printer) 或顯示器 (CRT display) 是最常用的周邊設備。其他如，磁碟 (Disk) 機、磁帶 (Tape) 機、繪圖機 (Plotter)、讀卡機 (Card reader) 及讀紙帶機 (Tape reader) 等。按各自使用目的及需要而選用。

軟體指與設備有別的程式 (Program)、語言 (Language)、常式 (Routine)、碼 (Code) 及其他寫出的資訊等。屬於使用法的部份用來操作計算機，以發揮它的功能。有很多學者將圖表 (Diagram)，要求敘述 (Requirement statement)，發展計劃 (Development schedules)，使用者手冊 (User's manual) 及其他配合發展一組程式的副產品也包括在軟體內。(注意：軟體和程式是有區別的，程式是屬於軟體之內。軟體表示相關的及互相作用的一群程式，而程式代表的意義小很多，它指執行一完整功能的一批密切關聯的指令表)。

第三節 數系、位元、字元

前面提到電子計算機像人腦一樣，具體地說明它的功能不外：(1) 輸入，(2) 儲存，(3) 控制，(4) 處理，(5) 輸出等五項。它的計算速度高達十億分之一秒 (nanosecond 簡寫為 ns. $= 10^{-9}$)，以這樣高的速率執行複雜的運算，電腦是使用二進位 (Binary) 形式來進行的。換句話說，



計算機系統

電腦只懂得以 0 和 1 組成的命令和資料。

二進位系統

二進位系統和十進位系統不同，它的計數法只有從零到一。增加數目就必須進位到一個新的位置。所以，它只使用 0 和 1 兩個符號。現在將十進數和相同數值用二進位寫法表列比較如下：

十進位數	二進位數	二進數的意義
0	0	零個 1
1 (2^0)	1	一個 1
2 (2^1)	10	一個 2 + 零個 1
3	11	一個 2 + 一個 1
4 (2^2)	100	一個 4 + 零個 2 + 零個 1
5	101	一個 4 + 零個 2 + 一個 1
6	110	一個 4 + 一個 2 + 零個 1
7	111	一個 4 + 一個 2 + 一個 1
8 (2^3)	1000	一個 8 + 零個 4 + 零個 2 + 零個 1
9	1001	一個 8 + 零個 4 + 零個 2 + 一個 1
10	1010	一個 8 + 零個 4 + 一個 2 + 零個 1

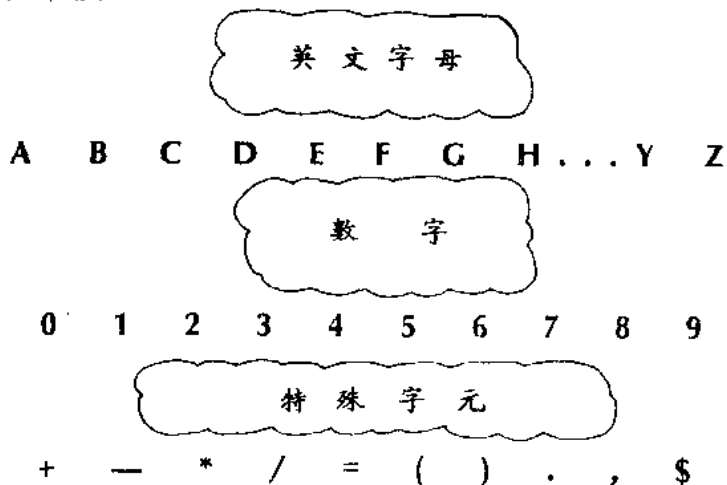
雖然，二進位算術用了較多的數目位置，但 0 和 1 兩個符號很容易轉換成電脈衝 (Electric impulse)，使電腦能夠接受而執行計算。

二進制的數基 (Radix) 為 2。在計算機裏儲存一個基數的記憶位置稱為位元 (Bit，又稱為數元。是 Binary digit 的縮寫)。是儲存裝置存放資料的單位，也是系統中資訊的最小部份。也就是說，一個位元的記憶電子線路或結構，能夠存放一個二進位基數，其內容不是 0，便是 1。

幾個位元組合起來，便構成位元組 (Byte，又稱為數元組)。或字組 (Word)。常見的是 8 位元的位元組，表示一個字元 (Character)。一個字組等於 16、24、32 位元等，隨機型而定。位元組在計算機主儲存器內當作可定址的 (Addressable) 記憶單位。由於電腦的記憶容量很大，所以我們常用 K byte 來表示記憶容量。1 K = 1024，例如，我們說 64 K bytes 即 $1024 \text{ byte} \times 64 = 65536$ 個位元組。(注意：K 這個符號在一般的物理量中代表仟 (Kilo)，但在電腦領域裏

不能認作仟。)

字元是指由計算機儲存或處理用的表示資訊的符號，數字，字母，或標點符號。每一字元都有兩種形式：(1)爲人所了解的形式（文字的），它包括 0 到 9 的十進位數字，A 到 Z 的英文字母，標點符號，以及其他格式與控制符號等；(2)計算機能了解的形式（碼），由一群二進位位元組成。



字元集合

八進位系統與十六進位系統

數目大的二進數寫起來很長，例如，十進數的 7896 寫成二進數是 1111011011000。因此，在實用上通常將它寫成八進數 (Octal) 或十六進數 (Hexadecimal)。八進數只用了八個數字：0, 1, 2, 3, 4, 5, 6, 7，即其數基爲 8。將二進數改爲八進數方法很簡單，只要由小數點（二進數的小數點）開始，向左右每次三個位元分爲一組，將各組分別換成對等的“十進數字”，得到的結果按順序由左到右寫下來，就是與該二進數的相當的八進數。例如，將上面的二進數轉換成八進數，（此一二進數是整數，所以只向左分。）

$$\begin{array}{cccccc}
 001 & 111 & 011 & 011 & 000 & \\
 1 & 7 & 3 & 3 & 0 & = 17330_8
 \end{array}$$

答案 17330_8 的右下角有一個很小的“8”字。在這裏它只是一個說明符號，這類符號稱為下標或腳註 (Subscript)。 17330_8 表示 17330 是 8 進位制的數目。

十六進數所使用的數字 (符號) 是：

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

二進數改寫成十六進數，完全和二進數改寫成八進數的原理相同。只不過每次是四個位元為一組，若一組內換成對等的十進數是大於 9 的數字，就分別以 A, B, C, D, E, F 來表示。例如，將二進數 1111011011000 改寫成十六進數如下：

$$\begin{array}{ccccccc} 0001 & 1110 & 1101 & 1000 & & & \\ 1 & 14 & 13 & 8 & & & \\ & E & D & & & & \end{array} = 1ED8_{16}$$

現在例舉零與最前面十六個整數的十進制，二進制，八進制，十六進制寫法對照如下：

十六進制	十進制	八進制	二進制
0	0	0	0000
1	1	1	0001
2	2	2	0010
3	3	3	0011
4	4	4	0100
5	5	5	0101
6	6	6	0110
7	7	7	0111
8	8	10	1000
9	9	11	1001
A	10	12	1010
B	11	13	1011
C	12	14	1100
D	13	15	1101
E	14	16	1110
F	15	17	1111
10	16	20	10000

至於由八進數、十六進數改寫成二進數，只要將上面的方法反過

來作。讀者可自行練習。

第四節 計算機運算程序

前面談過，電腦不能主動思考，也不會一看圖形或文字就能理解其意義。它不懂人類的語言，我們必須轉換成它能了解的語言（如 BASIC，FORTRAN 等）而接受指令，從事運算。

為了使讀者容易了解，現在將使用電腦及電腦的操作（運算）程序逐次說明如下：

1. 先設計程式，即程式規劃（Programming）。將要求電腦執行的事項以某種程式語言寫好程式。這也就是程式師（Programmer）的工作。
2. 將寫好的程式轉載於某種輸入設備能夠接受的媒體上。這種轉載就是將資訊利用打卡機（Punching machine）打成穿孔卡片（Punched card）或穿孔紙帶（Punched paper tape），或者將資訊轉錄在磁碟機上，或者直接由鍵盤打入電腦內，視機型及設備而定。
3. 如用打孔卡片或打孔紙帶或磁碟，則將它們按順序放進讀卡機或讀紙帶機或磁碟內，讀進其內容。
4. 資料輸入電腦後，就算寫的程式完全正確，它仍不能立即執行。它須先檢查此一程式並將之翻譯成爲機器語言（即由 0 和 1 組成）的程式，稱爲目的程式（Object program），而原來寫的程式則稱爲源程式或原始程式（Source program）。負責偵錯及翻譯的本身也是一種程式，稱爲編譯程式（Compiler, Interpreter），通常是由製造廠商供應。每種電腦語言各有其專用的編譯程式。如果源程式有邏輯上的錯誤，亦即，書寫不合語法規定時，編譯程式會逐一找出來，要求改正，否則無法執行。
5. 編譯成機器語言後，即可由中央處理單位執行實際的運算。
6. 中央處理單位將計算結果用電傳打字機，或印字機，或經由 CRT 輸出。

第五節 程式規劃的幾個關鍵字

要電腦工作須使用命令 (Command)，一個指令作一項工作。爲了使讀者在學習程式規劃 (Programming) 以前，對用 BASIC 語言寫程式的一些關鍵字 (Keyword) 有一認識，本節中先舉幾個作簡單介紹，使有一初步概念。以後各章的例題中再隨時說明。

PRINT

PRINT 是一個輸出指令 (Output command)。如果你要計算機寫出 "IT IS A COMPUTER"，就在終端機鍵盤上打

例一 10 PRINT "IT IS A CØMPUTER"
 20 END

計算機即會照你所指示的打出 (或在 CRT 上顯示出)

IT IS A CØMPUTER

假如你要它計算 23.4 乘以 91，就打

例二 10 PRINT 23.4 * 91
 20 END

計算機執行你的指令，打出或顯示出計算的結果後停止。

2129.4
DØNE

上面兩個例子，就是兩個完整的程式。當然，它們都很簡單。不過，我們已可利用來說明寫程式的一些順序：

- (1) 在程式中每一行 (Line，應該說是每一“列”) 開頭須用一個正整數，如上兩例中的 10 及 20，這個數字是賦予該一列的號數。由於在電腦中，我們送入很多指令，它如何記得或如何分辨呢？所以，就在指令前加上號數。電腦看到號數就知道將指令放在記憶器裏。以便於將來查尋和按順序執行。
- (2) 習慣上，一程式中相鄰的列的號數間隔爲 10，如 10, 20, 30, 40, …。因爲，將來修改程式時，修改的指令也必須要有列號 (