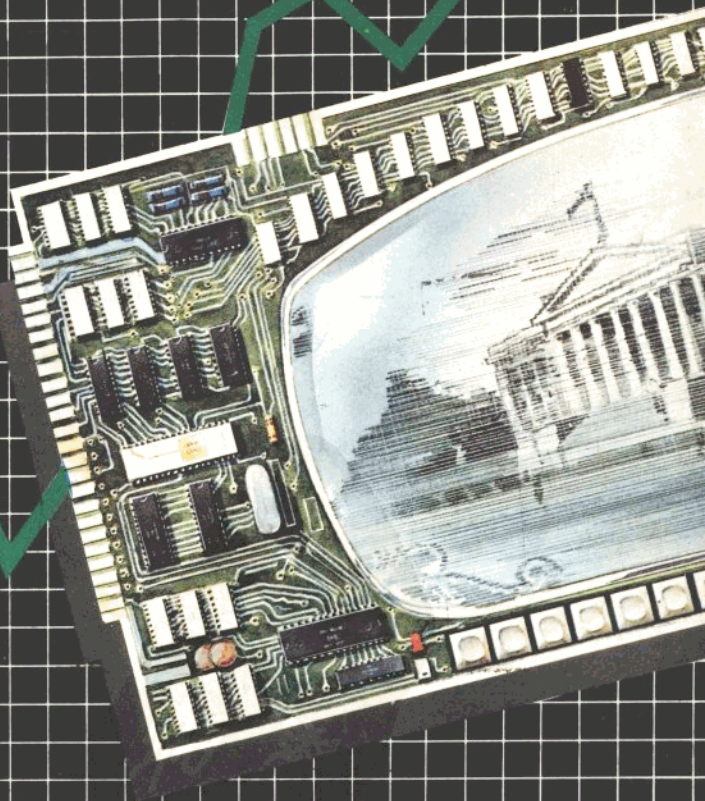


BASIC

在商業上的應用



周勝隣 譯

序 言

隨着計算機技術不斷的快速發展，以計算機來處理事務已經變成一個商務從業員必備的條件。也由於愈來愈多的商業事務電腦化——從文字處理到下定決心——每一階層的從業員都應該學習有關電腦的知識並了解此種“新機器”的強大功能及其一些限制。

學習電腦的最好方法或許就是由學習寫程式着手。本書就是以商務為應用的範疇，介紹 BASIC 的程式設計。BASIC 是一種相當簡單的程式語言，而且也是微電腦上應用最廣泛的語言。雖然 BASIC 很簡單易學，但其功能並不比其他語言遜色。

本書的目標是訓練您可以很快閱讀（追蹤）、設計並寫 BASIC 的商用程式，同時也教您如何去偵錯程式。在每一章中，我們針對 BASIC 的一種語法特性加以討論，而且至少以一個完整的商用程式詳細地說明該語法的功能及用途。

所有本書中所列的程式都有詳細的追蹤說明，使您能夠完全了解各程式的功能並熟悉程式設計的技巧。由於商用程式牽涉到商務會計上的一些知識，因此在很多程式之前，我們都先將程式所欲做的商務工作簡單地介紹說明，這樣子您更能很快地了解程式的工作性質，同時也讓您明瞭在設計程式之前如何將問題定義出來。

接著，我們來看一下每一章的內容概述：

第一章：BASIC 瀏覽 本章對程式概念作一通盤性的認識，同時也介紹本書中用到的一些辭彙。所討論的程式是一個銷貨成本程式，不過在此主要是讓您了解程式的組織大概，並不做深入的討論。

第二章：入門觀念 本章主要介紹 BASIC 的輸入及輸出，所討論的特性還有 PRINT USING 及 TAB 等，利用此等特性，我們就可以印製漂亮易讀的報表（我們亦將討論沒有此等特性如何去印製報表）。其他本章討論的課題還包括設定陳述、運算、判斷及控制轉換。所有這些，我們都將以一比較收入程式詳加說明與探討。

第三章：FOR 迴路 本章討論 BASIC 的重複執行的特性，也就如何不只一次地重複執行某段指令。所要討論的程式是兩種不同形式的月銷售報表程式，一種是只印簡單報表，另一個還加印長條圖。

第四章：陣列 本章探討在 BASIC 中如何將一組相關性的數據很有效地組織並儲存在所謂“陣列”的資料結構中。陣列是一種相當靈活好用的資料結構，我們將用計算現值的程式加以說明。

第五章：副程式及程式結構 本章利用簡化的分類帳程式一步一步地教您結構化程式的觀念。結構化程式是寫程式技巧中相當重要的一環。此章中就是教您如何寫一個漂亮的程式並如何去偵錯。由上而下及模組化等觀念都在討論之列。

第六章：算數函數 首先介紹固有函數如何應用到商務程式中，然後再介紹用者自定函數。有了用者自定函數的特性，您就可以定義一些系統沒有的算術函數了。本章所討論的程式有好幾個，不過較重要的是做成本數量利潤分析的損益平衡點程式。

第七章：字串 本章討論 BASIC 中有關於字串運算的一些函數，其中以兩個簡單的小程式來說明 ASCII 碼。另外，還介紹排序（將資料按某順序排列）的概念，並以一人員表列程式來說明。

本書中所列之程式都是在 TRS-80 或 Apple II 電腦上測試開發出來的。不過，我們隨時都有將一些會隨着 BASIC 版本之不同而

有差異的特性提示出來，並教您去測試看看此等差異。因此，研讀本書最好的方法就是要坐在您的電腦前，隨時將本書的程式執行看看。

每一章的最後都附有習題提供您更多練習的機會。附錄 A 則附有部分習題的解答供您參考。

本書另外一個特色就是在每一章後面還比較了其他三種語言 (COBOL、Pascal 及 FORTRAN) 的相關特性，這樣不僅讓您了解 BASIC 的優缺點，同時也拓展您認識的領域，使您對程式語言有更深入的了解。附錄 B 中，我們將每一章對其他三種語言所介紹的特性做一總結，分別討論以各種語言寫成的完整程式。雖然我們對其他三種語言的介紹只是皮毛，不過倒是可以讓您再要繼續學習上有所參考憑藉。

目 錄

序 言	V
第一章 BASIC之瀏覽	1
第一個BASIC程式：Cost of Goods Sold 程式.....	1
REM：BASIC 程式之註解.....	5
BASIC 的輸入及輸出.....	6
BASIC 之版本.....	7
執行COGS 程式.....	9
計算方法，流程圖及控制結構.....	10
輸出格式.....	16
其他語言：COBOL, Pascal 及 FORTRAN	16
結論.....	17
習題.....	18
第二章 入 門 觀 念	21
設定陳述.....	22
變數型態.....	23
算術運算.....	24
INPUT陳述	26
PRINT陳述	28
判斷，邏輯，及控制之轉移.....	33
比較損益表程式.....	43
其他語言.....	51

結論	55
習題	56
第三章 FOR迴路	61
迴路之重複次數	62
印製銷貨報表	64
巢狀迴路	67
印製長條圖	69
其他語言的迴路	74
結論	75
習題	76
第四章 陣 列	81
現值表	83
計算折扣因素之 BASIC 程式	86
陣列	88
改善的年金程式	90
折舊方法	93
幾個陣列範例	94
折舊現值	100
多因次陣列	107
其他語言之陣列	111
結論	112
習題	112
第五章 副程式及程式結構	115
分類帳	116

BASIC 的副程式語法	128
程式結構	130
一步一步地設計程式	132
結構良好之程式的偵錯	151
其他語言	152
結論	157
習題	157
第六章 算術函數	161
固有函數	161
亂數的統計分析	163
用者自定函數	167
成本數量利潤分析	172
具繪圖之損益平衡點程式	174
其他語言	182
結論	184
習題	185
第七章 字 串	189
ASCII 碼	189
說明 ASCII 的兩個程式	190
其他字串函數	194
其他語言	202
結論	202
習題	202

附錄 A 題解	205
附錄 B COBOL,Pascal及FORTRAN之完整程式 範例	213
參考書目	237

第一章

BASIC 之瀏覽

(A First Look at BASIC)

學習任何一種語言，我們都要學習兩類不同的辭彙。一類是關於程式語言本身，其中不只包含專用辭彙同時也牽涉到其語法及結構；另外一類則是在設計程式解決問題的過程中會碰到的一些辭彙。在此章中，我們將利用一個程式對於 BASIC 做一簡單，廣泛的了解，同時學習一些以後將會碰到的辭彙，以便能順利進入本書往後幾章之學習領域。

第一個 BASIC 程式：Cost of Goods Sold 程式 (COGS 銷貨成本)

本章中所討論之程式，其執行之工作可概分為下述三項：

- 提示用者鍵入程式所需之輸入資料，包括存貨，銷售等；
- 計算銷貨成本 (COGS, cost of goods sold) 以及銷貨毛利；
- 最後，將輸入之資料以及計算之結果以漂亮的格式印出來。

雖然此程式相當簡單，不過却可提供很多重要的觀念。追蹤此程式，我們可以初步了解一個 BASIC 程式如何設計以及書寫；同時相對應之流程圖更可以幫助我們將語言之控制結構（control structure）觀念具體表示出來。最後我們檢查輸出的結果，可以發現其報表格式經過精心設計，讓人一目了然。這種輸出報表的設計對於商業用途相當重要，因此如何設計一個清楚，漂亮而吸引人的輸出格式就是本書以後討論的一個課題。

圖 1-1 所示即為 COGS 程式。檢視此程式，我們注意到的第一件事是 BASIC 程式的每一行都有一行數，而這些行數的安排是由小到大。在 COGS 程式中，每一行數都是 10 的倍數，不過這並不是一定要如此，我們可以隨便選定，5 或 50 或 100 的倍數，甚至各行數不等距亦可。唯一的要求就是行數的編排要由小到大。雖然我們說行數可以任意選定，不過經驗告訴我們最好行與行之間留些“空間”以便需要時能插入新的指令。因為我們常為了某種需要得在原來程式中加入一些指令。假如我們將 COGS 程式之行數編為 1 到 65，那麼我們將無法在程式任何地方再加入新指令。

從 COGS 程式中，我們發現的另一個現象就是幾乎每一行都是以 REM, INPUT 或 PRINT 開頭。這三者都是 BASIC 語言之基本辭彙，亦即 BASIC 之保留字（reserved word）的一部分。INPUT 與 PRINT 主要是執行程式之輸入及輸出的工作，而 REM 則是用來註解程式。

```

10  REM      COGS PROGRAM
20  REM      D. HERGERT      8 AUGUST 1981
30  REM
40  REM      VARIABLE NAMES
50  REM      I                INCOME ON SALES
60  REM      B                BEGINNING INVENTORY
70  REM      P                PURCHASES DURING PERIOD
80  REM      E                ENDING INVENTORY
90  REM      C                COST OF GOODS SOLD
100 REM      G                GROSS MARGIN ON SALES
110 REM      A$              ANSWER STRING
120 REM
130 REM      INPUT SECTION
140 REM
150 INPUT "INCOME ON SALES"; I
160 INPUT "BEGINNING INVENTORY"; B
170 INPUT "PURCHASES DURING PERIOD"; P
180 INPUT "ENDING INVENTORY"; E
190 REM
200 REM      CALCULATION OF COGS AND GROSS MARGIN
210 REM
220 C = B + P - E
230 G = I - C
240 REM
250 REM      OUTPUT SECTION ( 輸出 )
260 REM
270 F$ = "$$##,####.##"
280 PRINT : PRINT
290 PRINT TAB(30); "COGS" : PRINT : PRINT

```

圖 1-1：銷貨成本程式 (COGS)

```

300 PRINT "INCOME ON SALES"; TAB(42);
310 PRINT USING F$; I
320 GOSUB 600
330 PRINT "BEGINNING INVENTORY"; TAB(30);
340 PRINT USING F$; B
350 PRINT "PURCHASES DURING PERIOD"; TAB(30);
360 PRINT USING F$; P
370 GOSUB 600
380 PRINT "GOODS AVAILABLE FOR SALE"; TAB(30);
390 PRINT USING F$; B + P
400 PRINT "ENDING INVENTORY"; TAB(30);
410 PRINT USING F$; E
420 GOSUB 600
430 PRINT "COST OF GOODS SOLD"; TAB(42);
440 PRINT USING F$; C
450 PRINT "GROSS MARGIN ON SALES"; TAB(42);
460 PRINT USING F$; G
470 GOSUB 630
480 INPUT "ANOTHER SET OF DATA"; A$
490 IF (A$="Y") OR (A$="YES") GOTO 150
500 STOP
600 PRINT TAB(30); "-----"
610 PRINT
620 RETURN
630 PRINT TAB(30); "-----"
640 PRINT
650 RETURN

```

圖 1-1 (續)

REM：BASIC程式之註解

程式中以REM開頭的每一行並非寫程式者給電腦的指令而是用來幫助他人了解程式之用（甚至，程式的原作者在經過一段長時間之後也需靠這些註解才能很快了解程式）。在REM行中，可以寫任何東西，因為註解說明一個程式並無一個標準方式。雖然BASIC程式並非很難了解，不過它一些特性也可能造成混淆，因此一個優秀的程式師應該知道程式中之困難特性而利用REM加以註解。這樣子別人就可以容易地了解程式之結構。

COGS程式中有幾處的REM行。第一處（10～20行）是說明程式的名字，作者以及完成的日期。一個程式過些日子可能需要作些修正，因此會有好幾個不同之版本，因此程式完成之日期通常是一相當重要的資料，如此擁有程式者才能分辨出該程式之版本。

第二處REM行（40～110行）則是對程式中所用到之變數（variable）作一說明。關於變數在本章稍後會再提及，暫時我們只需知道變數是程式中用來儲存數值的。每一個變數都有一個名字，不過在大部分BASIC版本中，變數名字的长度都限制於2或3個字。這點與其他電腦語言有很大差別，其他電腦允許較長的變數名字，因此變數名字可以取具有其代表之意義，如：

```
NET-INC
NET_INCOME
NINCOME
```

而在BASIC中，我們取變數名字似乎被宥於一個文字如：

```
N
```

或者是一個文字之後再加一個數目字，如：

N1

正因為如此，程式中利用 REM 將其使用到之變數列出來並加以簡單說明可以說深具價值。

最後讓我們來看看將程式區分為輸入區段 (INPUT SECTION)，計算銷貨成本及邊際毛利 (CALCULATION OF COGS AND GROSS MARGIN)，以及輸出區段 (OUTPUT SECTION) 之 REM。在其他高階語言，區段 (Section)，副程式 (Routine) 或是區塊 (Block) 可能代表某種特定的意義，但在 BASIC 中都只是代表程式中執行某一特定工作的一段程式指令而已。因此，REM 就用來將程式作註解，將之區分為不同的區段，使程式結構更為清楚。

總之，REM 之使用是根據他人閱讀程式時所會遭遇到的困難而定，因此程式師必需要能判斷何處須要加 REM 註解最為合適。記得，REM 雖非執行指令，不過仍佔電腦之記憶體空間，這對於記憶體有限而程式又很長時可以說是一大缺憾。雖然如此，我還是要強調在 BASIC 程式中 REM 註解仍是很重要的一環。

BASIC的輸入及輸出 (BASIC Input/Output)

BASIC 是一種編譯執行式 (interactive ，或稱交談式) 的程式語言，亦即在程式執行時我們可以從鍵盤鍵入程式所需之資料。換句話說，當程式在執行時，用者透過程式中某些指令可以跟電腦對話。其他不是編譯執行式語言，通常在程式執行之前就得將所有需要的輸入資料輸進去。這種程式執行的方法稱為成批式 (batch mode)。在本書中每章之末節所提到之 FORTRAN 及 COBOL 兩種語言原來之設計就是這種成批式執行的電腦語言。

BASIC 中構成用者與電腦交談的主要指令是 INPUT 及 OUTPUT。這兩者我們在第二章將會詳細討論，在此我們先檢視 COGS 程式中這兩者之作用即可。COGS 中 150 行到 180 行都是 INPUT，我們馬上可以看到，這每一行都會印出一些提詞告訴用者該輸入何種數據給電腦。例如：

```
BEGINNING INVENTORY?
```

(期初之存貨？)

BEGINNING INVENTORY 這句提詞就告訴用者在此時就應輸入原來之存貨的資料。大部分版本的 BASIC 都會在此時印出一個問號 (?) 以便讓用者知道程式正等待着輸入。

PRINT 是另外一個構成 BASIC 能夠交談特性的指令，其功能是印出欲輸出之結果。依照電腦所連結的週邊裝置（電視螢幕或列表機），PRINT 可以將輸出顯示在電視螢幕或印到列表機。

一行指令中可以只含 PRINT 一個字，如此在輸出結果上會產生一空白行。例如 COGS 程式中之 280 行

```
280 PRINT : PRINT
```

支點：在 BASIC 中是用來分開寫在同一行的兩個指令（一些版本的 BASIC 可能是用 \ (backslash) 而非支點）。因為如此，280 行將產生兩行之空白。

只要執行 COGS 程式，我們就可稍微了解人腦之交談並可以清楚 PRINT 及 INPUT 之真實功用。不過在還沒執行該程式之前還是讓我們先弄清楚一些重要的術語及其意義。

BASIC之版本 (Version of BASIC)

到現在為止我們已經不只一次提到 BASIC 之不同版本。因為我們馬上就要執行第一個 BASIC 程式，正好將 BASIC 之不同版本作一了解。當然如此一來，COGS 程式之執行又要耽擱一會兒了。

跟 FORTRAN, COBOL 及 PASCAL 等程式語言一樣，BASIC 也是一種高階 (high - level) 程式語言，亦即其指令都以英文表示，與人類之自然習慣相當吻合。不過電腦本身是不認識這些英文字的，它只認得一串串 0 與 1 組成的二進碼 (binary code)，而電腦能夠執行的其實是一些 0 與 1 組成代表程式指令意思的機器碼 (machine code)。但是，這樣電腦將如何執行一個以英文為主之高階語言 (如 BASIC) 寫成的程式呢？這就要靠一個程式將此鴻溝連起來了。此種連結高階語言與機器語言之程式稱為編譯器 (compiler) 或解譯器 (interpreter)。編譯器與解譯器的差別在於其如何將翻譯後之機器碼交給電腦來執行。

編譯器的作法是將整個程式從頭至尾 (若沒有錯誤的話) 一次地轉換成電腦能執行之機器語言；而解譯器的作法則是每一次只針對程式之一行作轉換，然後再下一行，如此直到碰到錯誤 (程式執行將停於此) 或程式結束。

BASIC 原來之設計是一種編譯執行式的語言，因此大部份版本之 BASIC 也都是解譯器。因為一種程式語言之特性完全取決於它的編譯器 (或解譯器)，因此我們談不同版本之 BASIC，事實上就是其解譯器之不同。在我們學習 BASIC 的過程中，會常常發現一些特性是取決於解譯器，亦即不同版本之 BASIC 提供之特性並不一樣。

討論了不少有關於編譯器，解譯器以及機器語言等，讓我們再回到 COGS 程式。

執行COGS程式 (Running the COGS Program)

圖 1-2 所示為 COGS 程式執行所顯示之簡單交談及其輸出結果。我們可以看到在輸入之交談與輸出結果之間被空白隔開，這是前段所討論之 280 行之結果。在輸入之交談中包含四句提詞 (prompt) 以及四組回答此些問題之資料 (回答資料是由用者從終端機之鍵盤鍵入)。電腦將這些回答之資料存在程式中之變數內，如此在執行以下之指令，這些變數即代表該等值。例如，當程式顯示如下之提詞：

BEGINNING INVENTORY?

(期初之存貨？)

而你鍵入 7800，那麼程式就將此值存入變數 B 中，而往下之指令若用到 B，其代表之值即為 7800 (當然，若再存新的值到 B，那麼其代表之值就變了！)

在輸入交談之後印的是程式之結果 (見圖 1-2)。我們發現在輸入交談中輸入之資料亦顯示於輸出之結果中，這是一種良好的習慣，因為如此可以讓人家一下子就明瞭執行此程式所輸之值。

輸出結果中同時顯示了三項非輸入之值：GOODS AVAILABLE FOR SALE (可供銷售貨品)，COST OF GOODS SOLD (銷貨成本)，以及 GROSS MARGIN ON SALES (銷售之毛利)。這些項目是電腦利用輸入之值根據其計算方法計算出來的。在圖 1-2 之輸出結果中，讓我們看看它的輸出格式 (format)。每一行金錢數目之值都以小數點對整齊，而且在適當位置亦標出逗號 (,) 以及錢號 (\$)。不同種類之陳述之間以虛線隔開，使表格看來一目了然。當然，這麼漂亮而且清楚的輸出表格絕非偶然或自然形成的，這都是寫程式者在寫程式時要精心設計的。在第二章我們將討論設計商用報