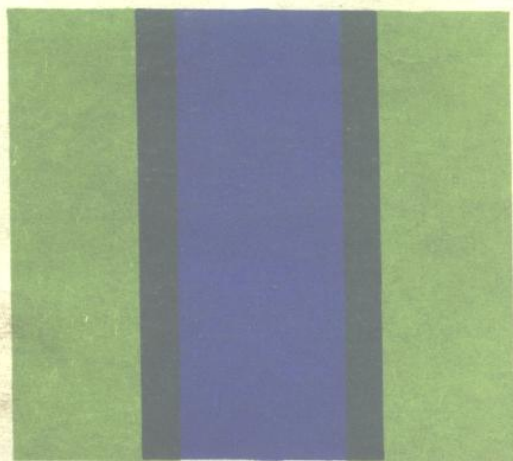




万用

BASIC手冊

周敏生 编著



台湾小教授电脑股份有限公司

版權所有



翻印必究

中華民國七十二年六月二十七日初版

萬用BASIC手冊

發行者：施振榮

著作者：周敏生

編輯者：姚芝芳

發行所：小教授電腦股份有限公司

台北市復興北路315號

郵政劃撥帳號：551262「第三波雜誌社」收

印刷者：沈氏藝術印刷股份有限公司

台北縣中和市中山路二段421號

電話：9556160～5

本出版社經行政院新聞局核准登記

登記證為局版台業字第2678號

如有缺頁空白倒裝，請寄回本社更換

全國各大書局有售

定價新台幣 100 元

73-87221

359

JS98/19

目

錄

如何使用本書.....	1	INPUT 指述.....	26
緒論.....	3	GO TO 指述.....	29
BASIC 語言.....	3	IF.....THEN 指述.....	30
程式範例.....	3	FOR.....NEXT 指述.....	32
語言要素.....	8	STOP 指述.....	35
數值常數.....	8	ON.....GO TO 指述.....	37
文字常數.....	9	副程式.....	37
數值變數.....	9	GO SUB 指述.....	38
文字變數.....	10	RETURN 指述.....	39
列變數.....	10	陣列: DIM 指述.....	40
數值陳式 (expression).....	11	函數.....	48
文字陳式.....	12	基本庫儲函數.....	49
關係陳式.....	13	三角函數.....	49
程式結構.....	13	指數函數.....	49
指述結構.....	14	算術函數.....	49
行號.....	14	實用函數.....	51
BASIC 指令 (Command)		RANDOMIZE 指述.....	52
LET 指述 (Statement).....	16	TAB 函數.....	53
PRINT 指述.....	17	使用者定義函數.....	55
END 指述.....	20	DEF 指述.....	55
REM 指述.....	21	資料檔.....	59
READ 指述.....	22	矩陣指述.....	65
DATA 指述.....	23	範例程式選粹.....	68
RESTORE 指述.....	25		

如何使用本書

本書的目的在於 BASIC 語言的整體介紹。在編寫時我們假設讀者沒有 BASIC 程式的知識和經驗，但對於已經熟識 BASIC 的讀者，由本書袖珍型的設計，也可以得到方便查考的便利。

本書在詳細討論各種不同的結構之前，都先列出一個程式範例來說明語言的結構。另外，在解決問題的同時，本書也介紹流程圖，可以方便讀者了解利用計算機解決問題之前必須經過的思維程序和流程結構。

BASIC 程式語言的各種類型定義都會先做粗略的介紹，而後再以各種指令來清晰的敘述它的含義。我們並不期待讀者第一次閱讀後便能窺得全貌，而希望在繼續地閱讀完全書後能夠逐漸地熟悉和了解全般的意義。

構成 BASIC 程式語言的各種指述都會以程式範例的方式逐一介紹。這些範例程式的目的是用來說明指述的用法，而非主要用於解決問題，爲了幫助讀者加深印象及提高思考興趣，在各種結構介紹以前，都會先有一個範例。

各種指令及指述的介紹並非依字母順序或任意排列，指述的介紹都是按它們的目的加以彙集再一一介紹的。介紹的次序如下：

LET, PRINT 及 END : 能夠寫出最簡單程式的基本指令。

REM : 增加程式的可閱讀性。

READ, DATA, RESTORE, INPUT: 資料、數值的設定。

GO TO, IF.....THEN, FOR...NEXT, STOP,
ON.....GO TO, GO SUB, RETURN: 控制程式流
程。

DIM, OPTION BASE : 以資料列 (DATA AR-
RAY) 的形式更有效及便利地掌握大量資料。

函數, RANDOMIZE, TAB 及 DEF : 有效解決程
式工作的特殊形式。

檔案指述 (File Statement) 。

MAT : 矩陣的應用。

本書最後以幾個不另外加上解釋的範例程式來作為結
束, 用以幫助對介紹過的各種指令做全盤的複習。

周敏生 謹誌

1983 年 6 月於左營

緒論

BASIC 語言

BASIC 語言的全名是 *Beginners All-purpose Symbolic Instruction Code*。它的結構簡單，字彙也不複雜，極適合計算機系統的初學者使用。但是 BASIC 也具備了足以解決大部份應用的特性，由於它在符號上採用基本的數學記號使它能解決數學型的問題，而同時它對字元的處理也能有效掌握，因此對於文字資料方面的問題也能適應。另一方面，不論大型計算機或微計算機都具備 BASIC 能力，也是它受歡迎的原因。

現有的 BASIC 語言有許多種，它們之間或有一些差異存在。本書是根據標準的 ECMA-55 定義編寫，也就是俗稱的“Minimal BASIC”。本書的絕大部份都能和各種 BASIC 語言吻合。

程式範例

問題：計算在一次考試中及格和不及格的學生人數，資料包括及格標準和各學生的分數。

程式最基本的技巧便是將欲交給計算機處理的問題，加以分割成能夠獲致答案的小步驟。任何我們要計算機做的事都必須考慮到並以詞句逐項敘述。而所有的步驟或語句都必須按先後次序排列。我們在進行學習間所碰到的例子或問題，大部份都能以很直接的方式加以解決。但在實際的應用上，問題愈複雜，計劃性的思維便愈發重要。一種最普遍的幫助計劃性地解決問題的工具便是流程圖，計算

及格與不及格學生人數的流程圖繪於圖 1。

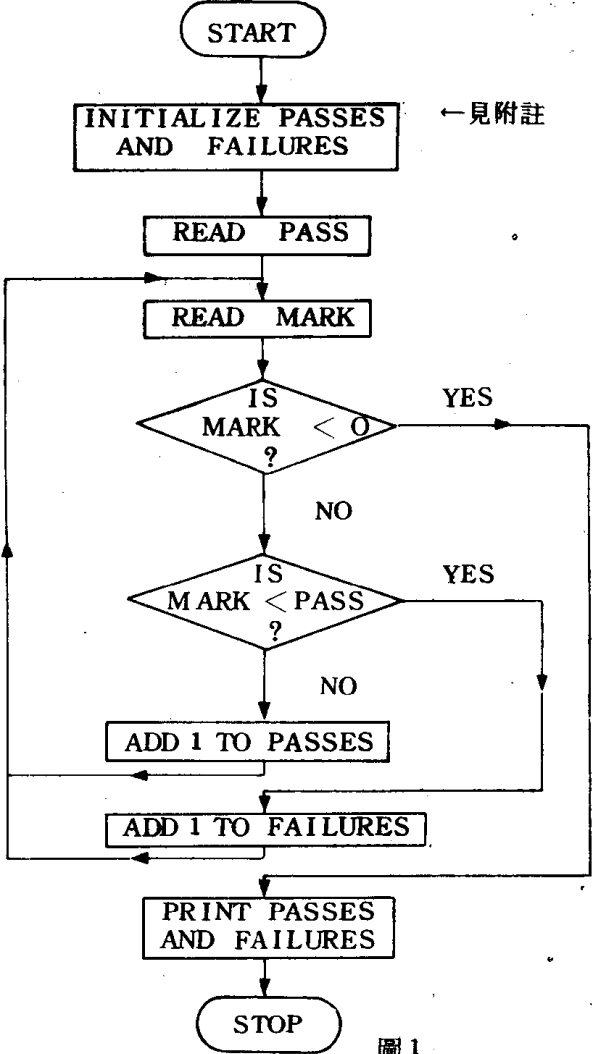


圖 1

附註：

- 1 方格代表執行某種“動作”。
- 2 菱形格子代表疑問的提出，有兩條流程綫由菱形格子流出，一個代表YES，一條代表NO。那個方向是YES沒有特別的規定，因此在格子旁都會附件YES和NO。
- 3  代表程式的開始和結束。
- 4 箭頭代表流程的方向。
- 5 流程圖幫助我們以簡單的方式敘述邏輯及發展解決問題的步驟，同時也幫助了解。
- 6 記住在循環的步驟開始以前及格和不及格人數都預先設定為0。

以下便是以 BASIC 語言寫出解決上述問題的程式。
本程式的輸出列於程式下方：

程式

```
010 REM*****
020 REM*EXAMPLE PROGRAM*
030 REM*FIND THE NUMBER OF CANDI-*
035 REM*DATES GAINING A PASS*
040 REM*IN AN EXAMINATION AND ALSO*
045 REM*COUNT FAILURES*
050 REM*
060 REM*P-NUMBER OF PASSES*
070 REM*F-NUMBER OF FAILURES*
080 REM*X-PASS MARK*
090 REM*M-CANDIDATE'S MARK*
100 REM*****
110 PRINT "EXAM ANALYSIS PROGRAM"
120 REM INITIALIZE PASS AND FAILURE
125 REM COUNTS
130 LET P=0
140 LET F=0
150 REM READ IN PASS MARK
160 READ X
170 REM READ IN NEXT MARK
180 READ M
190 REM EXIT FROM LOOP IF NO MORE
195 REM MARKS
```

```
200 IF M < 0 THEN 280
210 REM TEST MARK AGAINST PASS MARK
220 IF M < X THEN 250
230 P = P + 1
240 GO TO 180
250 F = F + 1
260 GO TO 180
270 REM END OF LOOP
280 PRINT P; " PASSES AND" ; F; " FAILURES"
290 DATA 45
300 DATA 56,75,41,80,67,30
310 DATA 17,48,58,83,24
320 DATA - 1
330 END
```

输出

```
EXAM ANALYSIS PROGRAM
    7 PASSES AND 4 FAILURES
```

語言要素

本章定義 BASIC 語言的各個組成部份，一方面也幫助讀者對以後將陸續介紹的各種指令的了解，同時亦避免在往後重複敘述。

BASIC 可以處理數值和文字資料。可以略分為數值常數和文字常數。

數值常數

BASIC 語言中的數值常數可以為整數，具有整數及小數點以下部份的實數或者是僅具小數部份的實數，以下是各種合法的形式。

7. -2. 12.43 .630 -0.07305 +2106418

符號省略時，該常數視為正數。注意不可有“，”符號。

BASIC 輸出數字時亦以類似的方式輸出，負數前帶一個負號，正數前則省略“+”號而替之以空白。同時，對於極大或極小的數字，BASIC 會自動以指數方式印出，亦即以科學符號表示，例如：

3.50000E+9	代替	3500000000
9.00000E-6		0.000009
-1.92000E-5		-0.0000192

數字亦可以指數型式輸入。

對於允許的有效位數和數值大小，各種計算機都有限制。Minimal Basic 定義的範圍由 $1E-38$ 至 $1E+38$ ，有效位數在六位以上。

文字常數

文字常數是指在 BASIC 中一串以引號括起的字元列。除了引號之外文字常數可為任何可印出字元，特別注意空白也是一個相當重要的字元形式。以下是字元常數的例子：

"MARY,MARY QUITE CONTARY"

"OCTOBER 1ST.1980"

" "

值得注意的是文字常數中可以包含數字字元，但是不能做任何數學的運算。

BASIC 同時也允許文字以 DATA 指述或相應於 INPUT 指述以鍵入的方式定義，而必須使用引號。

數值變數

變數係指示計算機中儲存位置的命名或符號，對於一個數值變數可以任意給定一個數值而且允許隨時改變。數值變數的命名可以為單一的英文字母或者是一個英文字母後跟隨一個阿拉伯數字形成。例如：

D P X1 N7

有些 BASIC 系統在程式執行前會自動將數值變數設定為 0。因此，在使用前未設定數值的變數會自動等於 0，由於並非所有的系統都有這種設定程序，在使用數值變數以前我們最好都能先設定數值。同時，有些系統允許兩個字母的數值變數。

文字變數

文字變數用以指出字元串列的儲存位址。基本的文字變數為一個英文字母之後加上金錢符號（\$），例如：

A\$ Q\$ C\$ Y\$

文字變數中可以容納的字元數依系統而有所不同，Minimal BASIC 規定最大長度至少在八個字元以上。

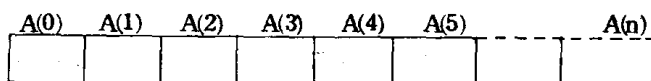
記住在計算機內部每個字元都以一個數字碼來代表。“A”的代表數字最小，而“Z”最大。在 BASIC 中，比較字元串列的大小即定比較各字元串列代表數字的大小。例如，X\$ 為“CAT”，Y\$ 為“DOG”，則 X\$ 值即小於 Y\$，因為 C 的代表數字小於 D。因此，在 BASIC 中文字排列的方式是很直覺的。

列變數（Array Variable）

在計算機程式中經常必須處理一些相關資料項的集合，對於這種集合給予它包含的資料項統一的名稱顯然較諸分別命名來得便利，列變數即是給予一個變數的集合統一的命名而後以註標方式來分別各個變數。列變數的命名法則類似於單一變數，但是數值列變數僅能以單一的字母來命名。註標係放在緊隨於變數命名後的括號中。由註標的出現，可以區別列變數和單一變數。含註標列變數的例子如：

A(5) D(N) B\$(3) L\$(K-1)

注意到註標可以變數的方式表示，若是列由 A(0) 開始，則 A(5) 表示列中的第六個數值變數。



有些系統第一個位置以A(1)表示，A(0)則不具意義。而有些系統則可以由程式員指定由0或1開始，在DIM指述中，會對此有更詳細的說明。

同時，在同一個程式中，不可用同一個字母來代替單一變數和列變數。

BASIC允許程式者利用DIM指述來宣告列的大小。如果沒有宣告，依系統的不同會自動給定註標位置由0到10或1到10。

通常BASIC亦可處理二維列，二維列中的元素以由逗號加以分開的一對註標來表示。

E(3,4) R\$(20,15)

第一個註標代表列，第二個代表行。因此，數值列變數E有3列4行，12個元素。

數值陳式(Numeric Expressions)

陳式係由一序列以特殊運算符號陳述的常數和變數，用以計算其運算的數值。最基本的形式即為僅包含單一變數或單一常數的陳式。

數值陳式由以下幾種算術符號架構而成

+ 加法 * 乘法

- 減法 / 除法

指數(乘方)註：有些系統以↑或^來代替號表示乘方。

在混合陳式中(含一種以上的運算符號)，運算的優

先次序和一般代數運算相同。乘方優先順序最高，其次為乘或除，依其在陳式中排列的順序。加和減則最後運算。

注意，括號可以改變優先順序，表 1 為代數運算順序的例子。

表 1

數值陳式	次序	運算	結果
$A+B \cdot C/D \cdot \cdot E$	1	$D \cdot \cdot E$	R1
	2	$B \cdot C$	R2
	3	$R2/R1$	R3
	4	$A+R3$	R4
R4 為陳式運算結果			
$(A+B) \cdot C/D \cdot \cdot E$	1	$A+B$	T1
	2	$D \cdot \cdot E$	T2
	3	$T1 \cdot C$	T3
	4	$T3/T2$	T4
T4 為陳式運算結果			

同時必須注意兩個代數運算不能緊鄰放置，例如 $X/-Y$ 正確的形式應該為 $X/(-Y)$ 。同時運算符號不可按一般習慣加以省略，例如 $A(B+1)$ 必須改為 $A \cdot (B+1)$ 。

文字陳式(String Expressions)

有些BASIC系統允許兩個文字串列的結合，稱之為文字連結運算。加號作為這種運算的運算子，例如 $X\$+Y\$$ 。

關係陳式 (Relational Expressions)

關係陳式由兩個數值陳式或兩個文字陳式所組成，中間再隔之以關係運算子。BASIC 有六種關係運算子。

運算子	代表意義
=	等於
<>	不等於
<	小於
<=	小於等於
>	大於
>=	大於等於

關係陳式僅用於 IF THEN 指述上。

有些 BASIC 系統允許使用邏輯運算子將兩個以上的關係陳式加以結合為複合關係陳式，邏輯運算子為 AND, OR 及 NOT。

程式結構

BASIC 程式是由一系列的指令以一連串的 BASIC 指述形式所構成。每個指述或指令都構成一行，各行有其指定的行號 (line number)。行號的作用有如標記一般，並且表示在執行時指述的組合順序。適當地以行號作為指示，我們可以將指述由儲存區中間取出加以修改。程式指述依照行號大小順序執行，除非碰到改變順序的指令，或者是在碰到 END 或 STOP 指述時終止執行。END 指述永遠是程式中最後的指述，因此也總是具有最大的行號。

指述結構

每個指述設定以前都必須先給它定一個行號，指出它在程式中的順序。行號之後，緊跟著是 BASIC 指令，說明將執行何種步驟，每個指述都放在各別的行數上，而且每一行只能有一個指述，因此指述的最大長度不能超過72個字元。行號必須為整數且介於 1 - 9999 之間，但有些系統範圍為 1 - 99999 。

指述中的合法間隔隨著系統的不同而不同，但一般來說，在各行的開頭，行號，指令都不要插入間隔。總之，根據指令的結構，適當的留出間隔可以增加程式的可讀性，一般來說是值得採用的。

行號

前文中已經指出，行號必須以介於 1-9999 的整數來表示。通常我們在編定行號時不使用連續的整數，而使相鄰的行號間留有間隔。因為程式往往須要經過多次的修改，行號留有間隔可以允許程式的修改或增減。通常行號的間隔為10。

BASIC 指令

在以下的篇幅中我們將一一介紹 BASIC 指令。介紹的方式是解釋指令的意義及文法上的規定，同時也有範例程式幫助說明指令和指述的使用。

每個程式範例都會列出組合出該程式的所有指令，按