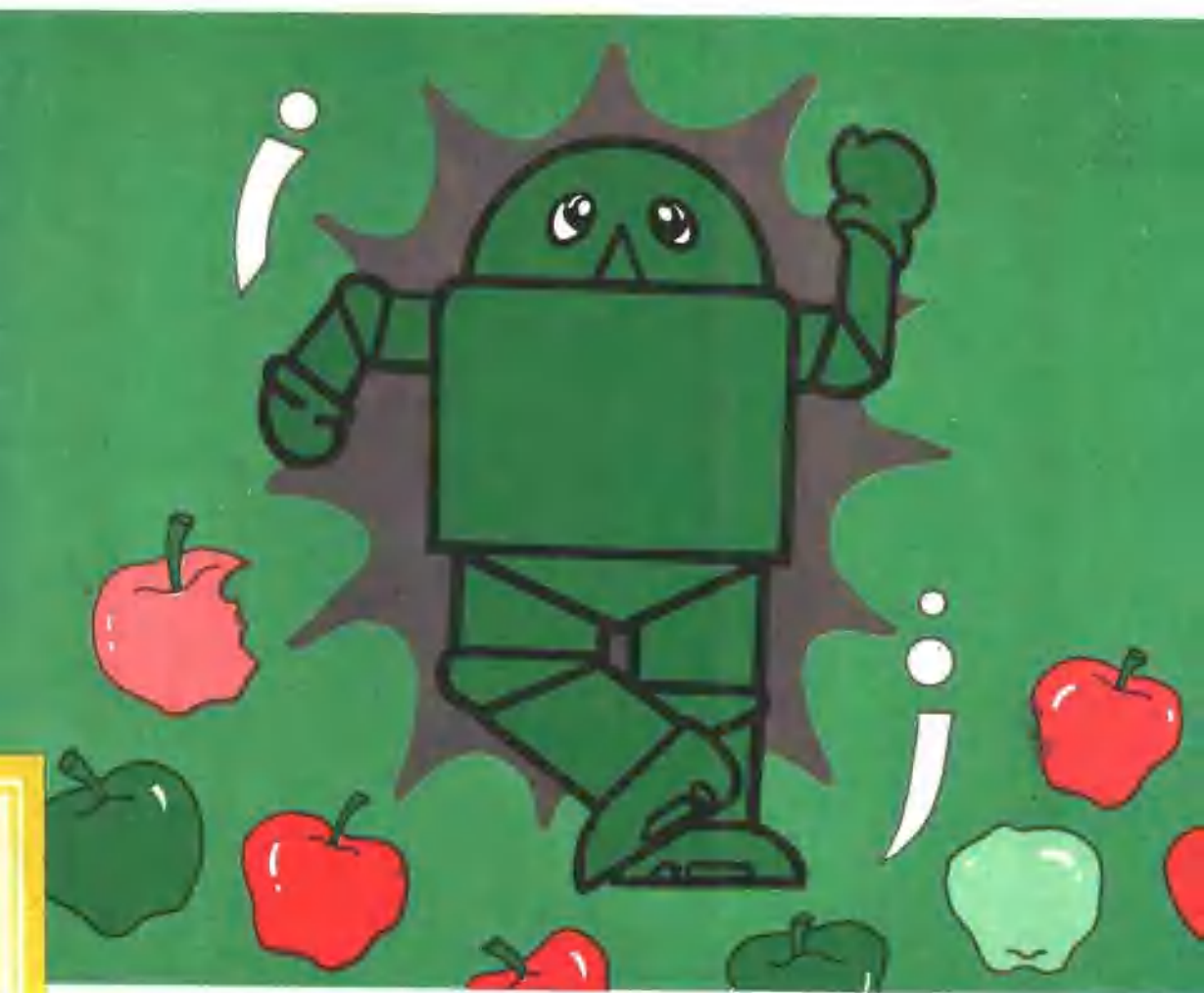


Apple II BASIC

電腦趣味遊戲

蔡富雄 編著

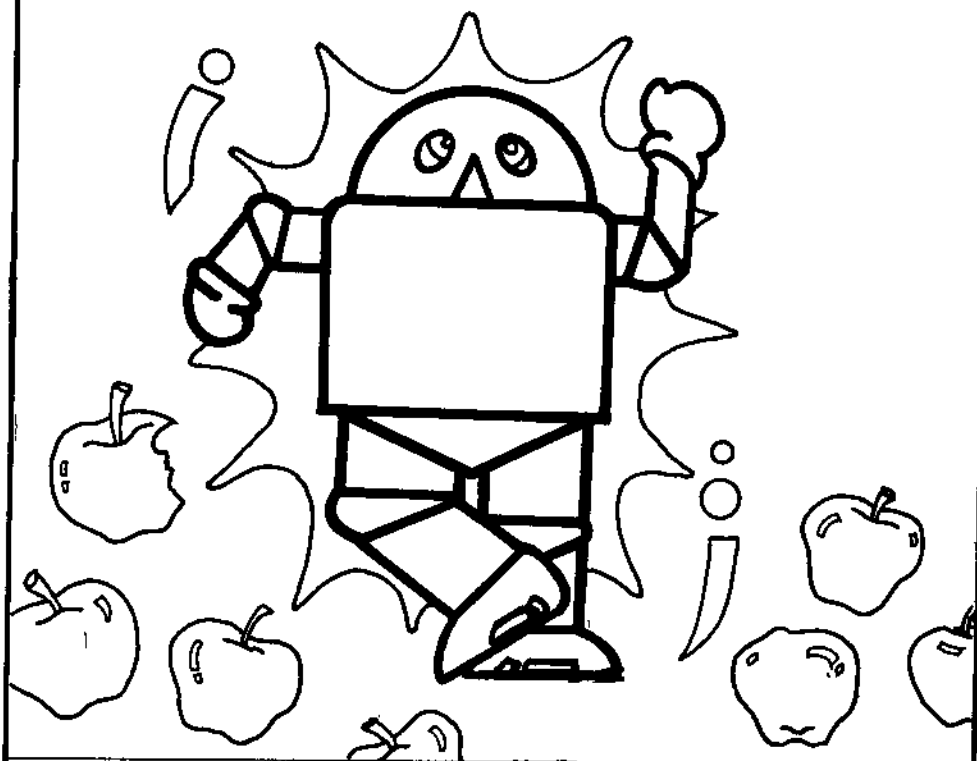


全華科技圖書股份有限公司 印行

Apple II BASIC

電腦趣味遊戲

蔡富雄 編著



全華科技圖書股份有限公司 印行



全華圖書 版權所有 翻印必究
局版台業字第0223號 法律顧問：陳培豪律師

APPLE II BASIC

電腦趣味遊戲

蔡富雄 編著

出版者 全華科技圖書股份有限公司
北市龍江路76巷20-2號
電話：581-1300・541-5342
581-1362・581-1347
郵政帳號：1 0 0 8 3 6

發行人 陳 本 源
印刷者 建華彩色印刷廠
定價 新臺幣 140 元
再版 中華民國72年10月



序

言

現在科技日益進步，學習電腦是大家所迫切需要的，而且電腦遊戲不但娛樂性非常之高，又富有益智的情趣，在遊戲之中，能夠啓發創造的意念，培養純熟的操作技能，並認識電腦的應用價值，使大家對電腦語言程式發生研究的興趣，進而體會出電腦結構的真善美。

本書共收集了二十多個遊戲程式，有最簡單的，也有非常複雜的，遊戲程式共分兩部份，第一部份是有關 Apple II 的一些遊戲程式，第二部份屬於任何微電腦 BASIC 語言程式遊戲，您可以從遊戲的過程，學習程式設計的方法與技巧，進而自己嘗試設計遊戲的程式，使您自己也會設計遊戲的程式，這就是本書編輯的最大目的。

本書共分三章，第一章把常用的 BASIC 語言的指令詳加介紹，使得您在研究程式時，使深切瞭解電腦執行的經過情形。第二章介紹程式的轉化方法，雖然遊戲的程式，是以 BASIC 語言寫的，然而各種微電腦都有它本身的特色，因此有些指令會有不同，而需要轉化才能執行，這裡我們告訴您如何來轉化。第三章是有關遊戲的程式及方法，分 Apple II 程式及一般 BASIC 語言程式，兩種遊戲程式。第四章是附錄，把遊戲程式要用到的一些知識加以介紹。

本書編輯期間，承蒙陳本源先生的鼓勵及鼎力襄助，和高雄市中正高工——劉

訓高老師的多方協助，實在非常感激，謹在此向他們致萬分之謝意。

本書編輯歷時久矣，然錯誤在所難免，希望先進不吝賜教。

蔡 富 雄 謹識

民國 72 年 3 月



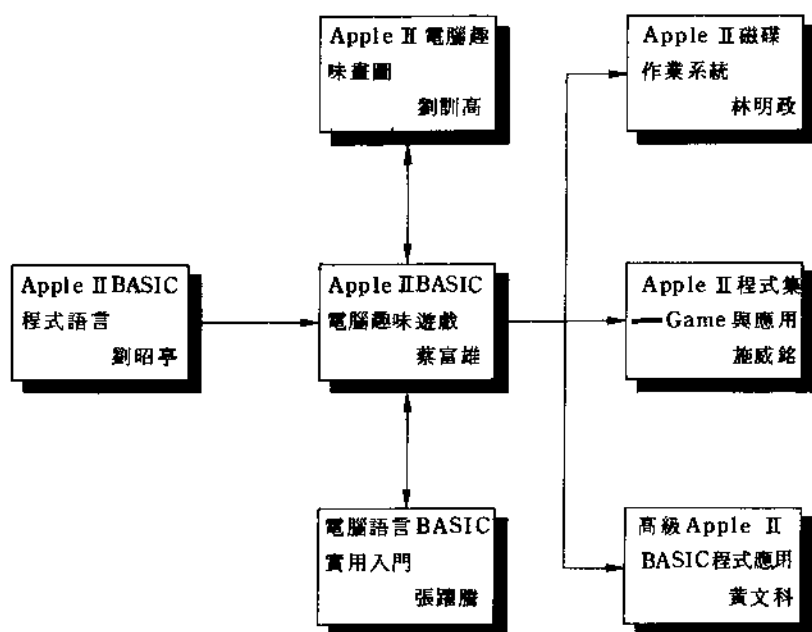
編輯部序

「系統編輯」是我們的編輯方針，我們所提供給您的，絕不只是一本書，而是關於這門學問的所有知識，它們由淺入深，循序漸進。

現在，我們將這本「APPLE II BASIC 電腦趣味遊戲」呈獻給您。玩過電腦遊戲的人；少數的感覺是很好玩、很過癮，而絕大多數不但有以上的感覺，而且會聯想到這些遊戲是如何被設計出來的？甚至想要看看這些遊戲程式的內容，並研究它的設計過程，期使自己有一天也能自行設計一些遊戲以自娛娛人。但是，若將一般在市面上發售的電腦遊戲軟體程式輸入，不是只能看到那些遊戲畫面，就是跑出一些莫名其妙的數字、符號（組合語言）。難道用 BASIC 語言就不能設計遊戲嗎？不！BASIC 程式只是因必須要經過翻譯，執行速度較慢而已，但一樣可以設計出一些趣味的遊戲，而市面上目前雖有一些電腦遊戲的書，但也是大都以艱深的組合語言介紹，或是說明不太詳細。本書針對大多數初學者的需要，提供了許多以 BASIC 語言所設計的遊戲程式，除供讀者輸入電腦後，用磁帶或磁碟儲存起來外，最主要的是，讀者可由研究這一些程式的設計過程、遊戲過程，體會出遊戲程式設計的方法和技巧，進而自行設計出一些簡單的遊戲程式。剛剛開始或許設計得不太精彩，但假以時日經驗豐富後，必能精益求精，相信有一天必能成為電腦趣味遊戲的設計師，不僅為自己帶來樂趣，培養嗜好，進而更為別人帶來快樂。

同時，為了使您能有系統且循序漸進研習 APPLE II BASIC 電腦趣味遊戲方面叢書，我們以流程圖方式，列出各有關圖書的閱讀順序，以減少您研習此門學問的摸索時間，並能對這門學問有完整的知識。若您在這方面有任何問題，歡迎來函連繫，我們將竭誠為您服務。

流程圖：



感謝您

感謝您選購全華圖書！

希望本書能滿足您求知的慾望！

圖書之可貴在其量也在其質

量指圖書內容充實、質指資料新穎够水準，我們就是本著這個原則，竭心盡力地為國家科學中文化努力，貢獻給您這一本全是精華的全華圖書。



第一章 BASIC 語言	1
第二章 程式轉化法	9
一、複指述	9
二、迴圈	10
三、陣列	11
四、字串符號	11
第三章 遊戲程式範例	17
3-1 Apple II 遊戲程式範例	17
3-1-1 唐老鴨	17
3-1-2 外星人	22
3-1-3 黑白棋	32
3-1-4 智慧盤	49
3-1-5 二十一點	57
3-1-6 登陸月球	64
3-1-7 兩張撲克牌	67
3-1-8 左輪手槍	69

3-2 BASIC 語言遊戲程式範例.....	70
3-2-1 俄式輪盤賭.....	70
3-2-2 五子棋.....	73
3-2-3 小白兔.....	76
3-2-4 SIN曲線.....	77
3-2-5 魔術方塊.....	81
3-2-6 跳 棋.....	85
3-2-7 猜數遊戲.....	91
3-2-8 畫方塊.....	95
3-2-9 數學骰子.....	99
3-2-10 取豆子遊戲.....	104
3-2-11 井字遊戲.....	113
3-2-12 彈回遊戲.....	118
3-2-13 黑盒子.....	122
3-2-14 圍捕遊戲.....	125
3-2-15 四子棋.....	129
3-2-16 西洋棋.....	135
3-2-17 搶黃金遊戲.....	145
3-2-18 擲骰子.....	154
3-2-19 畫標題.....	156
3-2-20 六子棋遊戲.....	160
3-2-21 畫 LOVE.....	167
附錄 A 編修用指令.....	171
附錄 B Apple II 固有的一些副程式.....	173
附錄 C 有用的 PEEK 及 POKE 位置	179
附錄 D Applesoft 保留字及 ASCII 碼.....	185
附錄 E 換算表.....	187



BASIC語言

(1) DATA——預存資料

格式：行號 DATA 數據 1，數據 2，……等

功用：

- ① DATA 預存固定不變的數值或變數值，以便READ按序讀取。
- ② DATA可放在程式的END 前的任何地方。但爲了方便通常都把 DATA 放在一起，或放在程式的最後面，以便於程式的修改。
- ③ 若READ中所需的變數比DATA多時，電腦出現“OUT OF DATA”的信息，程式將因此停止執行。

(2) READ——閱讀資料

格式：行號 READ 變數 1，變數 2，……等

功用：

- ① 在READ敘令中，逗點是用來區別數個不同的變數，而在DATA敘述裡，是用來分別各個變數所對應的數據。
- ② READ表示逐次讀取DATA中的數據，而DATA中所有的數據是供READ讀取的，依其行號之次序，自左而右，由上而下的逐一被讀取。

(3) DEF FNX(X)——定義函數

格式：行號 DEF FNa(X) = 算術運算式

功用：

- ① DEF FNa(X)中a表示英文字母A, B, C, ……Z中之一個字，也可為數目字0~9中任一數，而X為數值變數。
- ② FNa就是定義函數的函數名稱，X為虛設變數。
- ③ 例如：10 DEF FNA(X) = 3 * X ↑ 2 + 2 * X + 1

(4) DIM——度量敘令

格式：行號 DIM A(K₁), B(K₂), ……Z(K_n)

功用：

- ① 其中A, B, C, ……Z表示程式中所有各註標變數之行列名稱，而K₁, K₂, ……K_n各為一個或二個以上，以逗點分開，不帶正、負號且不為零之整常數，此常數表各註標之最大值。
- ② DIM通常放在程式的最前面。
- ③ DIM用以宣告行列的維數，給該行列預留了儲存位置，且說明該行列之維數及其元素的數目。

(5) END——結束敘令

格式：行號 END

功用：

- ① END表示程式的結束，故必須是最大的行號，或程式的最後一個敘述。
- ② 有些微電腦END敘令可以省略。

(6) FOR……TO……STEP——循環敘令

格式：行號 FOR I = n₁ TO n₂ STEP n₃

：

行號 NEXT I

功用：

- ① I表示任何數值變數（非字串變數）。
- ② n₁表示起始值，n₂表示終值，n₁及n₂可以是任何數字。
- ③ n₃表示增加值或減少值，故n₃可為正值或負值。
- ④ 當n₃ = 1時，STEP 1可以省略不寫，如果寫上了也沒錯。
- ⑤ 當指令執行時，先將I設定為n₁，而後執行FOR的下一個敘述，如果執行NEXT I時，電腦會將n₁加上n₃，再回到FOR的敘述中去比較I與n₂的值，若I較n₂為大，則結束循環，就繼續執行下一個行號。

⑥ 若 $I < n_2$ ，則執行FOR和NEXT間的循環敘述。

(7) GOTO——無條件轉向敘令

格式：行號 GOTO n

功用：

- ① 其中 n 表另一行號，但 n 不可為變數。
- ② 執行這個敘令時，會阻止正常的程式執行順序，而把控制轉移到另一個程式中的行號 n ，由行號 n 依序執行下去。
- ③ GOTO 跳到指定的行號 n 後，便不再回頭。

(8) GOSUB.....RETURN——副程式指令

格式：行號 GOSUB n

功用：

- ① n 表示另一行號。
- ② GOSUB 是電腦跳到指定的副程式執行，一直執行到 RETURN 指令，副程式才結束，再回到 GOSUB 的下一個敘述指令來執行。

(9) IF.....THEN——條件轉向敘令

格式：行號 IF A THEN n

功用：

- ① 其中 n 表示敘述行號，而 A 表示邏輯或關係運算式。
- ② 若 A 為真，則執行行號 n ，再按序往下執行；若 A 為假，就不執行行號 n ，而直接執行下一個行號。

(10) INPUT——輸入敘令

格式：行號 INPUT $n_1, n_2, n_3, \dots$

功用：

- ① 其中 $n_1, n_2, n_3, \dots$ 均表示變數，這些變數可以是數值或文字變數，各變數之間必須以逗點分開。
- ② 當程式執行這個敘述時，電腦最後會顯示問號“？”，並等候操作者打入資料，輸入之資料要與 INPUT 的變數相同，否則電腦會發出錯誤的訊息。
- ③ INPUT 中帶頭的文數字串只能一個，而且要以分號（；）與其他變數分開。
- ④ 當輸入的字串含有逗號（，）、分號（；）或前置空格時，該字串必須以雙引號括起，否則電腦會顯示錯誤的訊息。

(11) LET——指定敘令

4 Apple II BASIC 電腦趣味遊戲

格式 1：行號 LET 變數=常數，變數或運算式。

格式 2：行號 LET 變數="文字系列"或變數。

功用：

- ① 式中等號並非一般算術符號，而是表示將右邊運算式，計算結果求出值，然後把它存放在左邊變數的位置內。
- ② 表示文字系列取代左邊的變數位置。
- ③ 在許多電腦系統中，皆可將LET省略。

(12) ON.....GOTO——計值轉向敘令

格式：行號 ON I GOTO $n_1, n_2, n_3, \dots, n_k$

功用：

- ① 其中 $n_1, n_2, n_3, \dots, n_k$ 分別表示行號，而I表示數值變數，其值為 $1 \leq I \leq k$ 。
- ② 若 $I = 1$ 執行敘述轉移到行號 n_1 。
 $I = 2$ 執行敘述轉移到行號 n_2 。
 $I = 3 \dots I = k$ 以此類推。
- ③ ON.....GOTO發生流程轉換，則轉換以後，並不回到原來跳出的地方。

(13) ON.....GOSUB——計算值轉向副程式敘令

格式：行號 ON I GOSUB $n_1, n_2, n_3, \dots$

功用：

- ① ON.....GOSUB與ON.....GOTO很相似，不同的地方是ON.....GOSUB轉換到副程式中去執行，若遇到RETURN指令時，則跳回ON.....GOSUB的下一行號去執行。

(14) GOSUB.....RETURN——副程式指令

格式：GOSUB n.....RETURN

功用：

- ① 這個指令會將控制權轉到副程式敘述的第一個指令上，一直執行到RETURN的指令，再跳回到GOSUB的下一個敘述去執行。
- ② 在指令中，n表任何一個副程式開始的行號，當副程式的所有指令執行完時，RETURN放在副程式的最後一個敘述上，用來指示電腦立即回到GOSUB的下一個敘述去執行。

(15) PRINT——印出敘令

格式 1：行號 PRINT 變數（或運算式），變數（或運算式）

功用：

變數（或運算式）與變數（或運算式）間必須以逗點（,）分開。

格式 2：行號 PRINT " "

功用：

是將某文字常數列印出來，其引號 " " 內的字是所須之訊息。

格式 3：行號 PRINT

功用：

表空一行，並無任何訊息或輸出之數據要印，因而跳一行。

〔註〕：

① 程式：10 PRINT A,B,C,D

功用：要印的文字或數字，以逗點（,）分開，表示各資料間是區分列印，按每區以 16 個字分隔。

② 程式：10 PRINT A;B;C;D

功用：要印的資料以分號（;）分開，表示各資料間是緊接着印出，兩資料間僅以一個字為間隔。

格式 4：行號 PRINT TAB(X)

功用：

① 在一列中第 X 個空間開始列印資料。

② 其中 X 表示空 X 個字。

(16) REM——備註敘令

格式：行號 REM 任何文句

功用：

① 為了使使用者瞭解程式中，某一階段的内容，因而加註 REM。

② 此敘令對程式的執行並無任何影響，可置於程式的任何地方。

(17) RESTORE——重覆預存資料

格式：行號 RESTORE

功用：

① RESTORE 敘令之使用，必須是和閱讀（READ）資料（DATA）敘令相配合，其功用是使 DATA 所定之數據能再次為 READ 所運用。

② RESTORE 的功能是使下次的 READ 回到 DATA 所儲字的第一個資料讀起，也就是用來將資料指標指到資料區的開始位置。

(18) STOP——停止敘令

格式：行號 STOP

或行號 STOP “文字符號”

功用：

- ① 表示執行到此敘述時暫停，回到接受指令的行號，STOP 可放在程式中的任何位置，並出現任意多次。
- ② 程式中可以有數個STOP敘令，並可賦予不同文字符號，以指示該程式停止時之情況。
- ③ 若在此時鍵入CON或CONTINUE命令時，程式會執行STOP的下一個行號。

(19) ABS(X)——絕對值函數

$$\text{ABS}(X) = \begin{cases} X & \text{當 } X \geq 0 \\ -X & \text{當 } X < 0 \end{cases}$$

(20) ASC(X\$)

ASC(X\$)的結果係字串表示之第一字轉換成ASCII CODE。

(21) ATN(X)——反切函數

X為強度，計算 $\tan^{-1} X$ 之值。

(22) CHR\$(X)——將ASCII碼轉換成文字字串

其中X在0與255之間的任意ASCII碼值。

(23) COS(X)——餘弦函數值

X為強度，計算COS(X)之值

(24) EXP(X)——指數函數

計算X之指數 e^x 之值，其中X為任意實數。

(25) INT(X)——最大整數函數

若 $\text{INT}(X) = N$ ，則 $N \leq X < N + 1$ ，其中N為整數。

(26) LEFT\$(X\$, Y)——在字串X\$中，向左取Y個字元。

(27) LEN(X\$)——求出字串長度，X\$可為字串常數或字串變數。

LEN(X\$)值在0到255之間。

(28) LOG(X)——對數函數

其中X的值必為正數，LOG(X)值則為實數。

(29) MID\$(X\$, Y, Z)——在字串X\$中，由左邊第Y個字元起，取Z個字元。

若Z省略不寫時，表示由左邊第Y個字元起到最後一個字元全部取完。

(30) RND(X)——隨機值函數

- ① 其中X可為任意實數，隨機值函數RND(X)的值介於0與1之間的實數中，抽取任意的數據。
- ② 如果要隨機取0到9的十個數據，只要把RND(X)乘以10即可，即 $10 * \text{RND}(X)$ ，然後再用INT(X)函數就可得到。
- ③ RND(X)中之X是一個虛擬的表示，只要有數字即可，在產生任意數時是必須的，但是並不發生其他作用。
- ④ 在RND(X)中
 - (a) 當 $X > 0$ 時，表示RND(X)依序產生下一個亂數。
 - (b) 當 $X = 0$ 時，表示RND(X)重複上一個取出的亂數。
 - (c) 當 $X < 0$ 時，表示RND(X)回到該序列第一個亂數開始產生亂數。
 - (d) 若只寫RND(X)則與(a)同。

(31) RIGHT\$(X\$, Y) 在字串X\$中，第Y個字元起向右取所有字元。

(32) SGN(X)——符號函數

若X為輻數，則

$$\text{SGN}(X) = \begin{cases} 1 & \text{當 } X > 0 \\ 0 & \text{當 } X = 0 \\ -1 & \text{當 } X < 0 \end{cases}$$

(33) SIN(X)——正弦函數

X為強度，計算SIN(X)之值。

$$(34) \quad 2\pi \text{ 強} = 360 \text{ 度}, \pi \text{ 強} = 180 \text{ 度}, 1 \text{ 強} = \frac{180}{\pi} \text{ 度}$$

$$\pi / 180 \text{ 強} = 1 \text{ 度}, \text{其中 } \pi = 3.14159$$

(35) 把某數A，四捨五入之方法是：

$$S = \text{INT}(A + \text{SGN}(A) * 0.5)$$

(36) SQR(X)——開平方根函數

其中 $X \geq 0$ 與 $X \uparrow \frac{1}{2}$ 完全相同，但速度較快

(37) STR\$(X)\$——將X之數值常數或數值變數轉換成文數字串，轉換成字串後，