

BASIC

程式設計範例

增訂版

許貴濱 編著
蒲浩哲
宋惠亮

林武雄 校閱

松崗電腦圖書資料有限公司

BASIC 程式設計範例

增訂版

許 貴 濱 編 著
蒲 浩 哲
宋 惠 亮
林 武 雄 校 閱

松崗電腦圖書資料有限公司

BASIC程式設計範例

書號：210103

版權所有



翻印必究

每本定價 110 元整

編著者：許貴濱、蒲浩哲、宋惠亮

發行人：吳守信

發行所：道明出版社

台北市仁愛路二段一一〇號三樓

總經理：松崗電腦圖書資料有限公司

台北市仁愛路二段一一〇號三樓

電話：3930255 · 3930249

郵政劃撥：109030

印刷者：泉崗印刷設計股份有限公司

台北市仁愛路二段一一〇號三樓

電話：3930255 · 3930249

中華民國六十七年三月初版

中華民國七十年三月第四版

中華民國七十一年八月第六版

本出版社經行政院新聞局核准登記，

登記證號為局版台業字第一七二九號

校者序

BASIC 乃英文 Beginner All-purpose Symbolic Instruction Code 之縮寫，是一種特為初學者設計的多用途程式語言，其簡單易學又好用，因此廣被各界所愛用，校者曾修習此種語言多年，時下除了王安電腦公司出有一本“王安 2200 型電腦語言”之外，似乎已再找不出第 2 本較完整介紹“BASIC”語言的中文書，本書作者以實例為主，從建檔開始到報表的產生，將整個作業過程一一的介紹給讀者，其間並把設計程式時事先應做的各種考慮及有關的設計技巧很巧妙的道了出來。如果說讀了“王安 2200 型電腦語言”能使您對“BASIC”登堂，則再讀了本書之後一定能引您入室，作者能在百忙之中完成此書，其對“BASIC”之推展可謂不遺餘力，望我讀者善加利用。

林武雄 敬識

民國六十六年十月于高雄

前 言

本範本之編寫取一貫系統作業分爲各個範例自建檔至報表之印製，作一系列詳盡之介紹。

本書之讀者仍著重於曾修習過 BASIC LANGUAGE 熟習 BASIC STATEMENTS 但無實際設計程式經驗或有設計程式經驗，但不知如何著手建立電腦作業系統者。

於此作者非常感謝林榮俊、柯順基兩位先生的賜教，及姚善淑小姐的協助。

作者才疏識淺本書乃初次嘗試，故而戰戰兢兢，竭思盡慮編寫本範本，然漏誤之處猶恐難免，祈請前輩專家讀者不吝指正。

許 貴 濱 謹 識

民國六十六年九月

增訂版序

MINI COMPUTER 在中小企業界占著一席重要的地位，因而 BASIC 成爲普遍使用的語言。諺語中有句話——「工欲善其事，必先利其器」，是以熟悉該種語言及機器特性後，自然能運用自如。

本書藉再版之機會，將每則例題由淺入深漸次介紹各種指令的用法及建檔（含中文）方式。諒必讀者研讀後，對 BASIC 語言能有更確切的認識，進而得以十分臻熟地駕馭機器。

筆者才學疏淺，蒙 國立中興大學電腦中心主任鍾英明先生及松崗電腦圖書公司經理吳守信先生之提携與鼓勵，乃敢嘗試，得以爲學術界略盡棉薄之力，又蒙王安公司系統部諸位先生給予指導，謹此一併申謝。倉促之間完稿，謬誤之處，在所難免，尙祈先進們不吝賜正。

編者 謹識

民國七十年三月

目 錄

基礎篇

範例一：速度之比較	1
範例二：躍動的圖形	3
範例三：固定的資料	5
範例四：圖案之印製	7
範例四之二：圖案之印製	9
範例五：錯誤的輸入	11
範例六：排高矮	13
範例七：文字的內儲型式	15
範例八：無窮數列	17
範例九：統計圖	19
範例九之二：統計圖	21
範例十：活了幾天	23
範例十一：一次聯立方程式	25
範例十二：模擬骰子實驗	27
範例十三：TABLE 之習作	29
範例十三之二：TABLE 之習作	31
範例十四：串列資料之MATCH	33
範例十五：FIBONACCI 數字遊戲	35
範例十六：溫度轉換	37

應用篇

例一：建檔一	39
例一：建檔一之二	41
例二：建檔二	49
例二：建檔二之二	51
例三：建檔三	57
例四：建檔四	59
例五：腳接 (CONCATENATION)	61

例六：原式抄印.....	63
例七：改錯 (CORRECTION)	65
例八：搜尋 (SEARCH)	67
例九：分類 (SORT)	69
例九之二：分類 (SORT)—數筆相同資料之處理法.....	73
例九之三：分類 (SORT)—問卷統計.....	79
例九之四：分類 (SORT)—電腦閱卷評分.....	83
例十 a：更新 (UPDATING)	87
例十 b：直接更新法 (DAM, DIRECT ACCESS METHOD)	91
例十 c：索引順序更新法 (ISAM, INDEX SEQUENTIAL ACCESS METHOD) 要點說明	97
例十一：合併 (MERGE)	99
例十一：合併 (MERGE) 之二.....	101
例十二：修改 (MODIFY)	107
例十三：BA 觀念	111
例十四：報表 1 (REPORT)	113
例十五：報表 2 (REPORT)	117
例十六：磁帶 (TAPE CASSETTES)	121
例十六之二：磁帶 (TAPE)	123

範例一：速度之比較

題目

比較 FOR-NEXT 與 IF-GOTO 之速度。

涵意

FOR-NEXT 的結構專為處理內容為漸增或漸減的變數，實質上也可以用 IF-GOTO 的結構來完成相同的工作。不過，到底何者之速度為快呢？

思考

既然我們的目的只是比較 FOR-NEXT 與 IF-GOTO 的速度而已，因此其他的敘述儘量避免採用。我們分為 2 個程式，一個為 FOR-NEXT，另一為 IF-GOTO。分別建入電算機中測量其執行的時間。

程式

A. 測試 FOR-NEXT 之時間

```
10 FOR A=1 TO 100000 STEP 1
20 NEXT A
30 END
```

B. 測試 IF-GOTO 之時間

```
10 A = 1
20 IF A > 100000 THEN 30 : A = A + 1 : GOTO 20
30 END
```

討論

- (1) A 之截止值雖暫定為 100000，但可隨意加以更改。
- (2) FOR 敘述之後若不接 STEP 1，對速度是否有影響？可試測之。
- (3) 程式 B 的式 20 若分開為 3 個敘述，對速度有無影響？試測之。
- (4) 若要知道加 1 的速度為何，可在程式 A 中加入

```
15 B = A + 1
```

然後測試之。新程式所費時間減去原程式所費時間即得 100000 個加算的時間。

範例二：躍動的圖形

題目

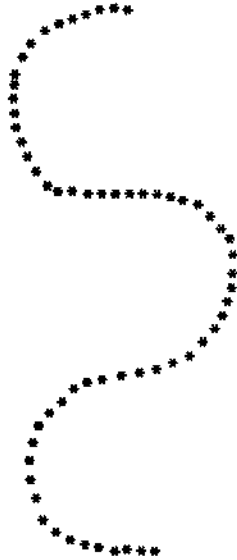
在螢幕上造成一波浪狀的曲線，不斷地擺動。

涵意

螢幕原本是一種表現動態圖畫的工具。躍動的圖形在電算機的應用上是一種很吸引人的表現形態，比起單調刻板的數字、文字要生動活潑許多。

思考

提起波浪狀的曲線，首先想到的是SIN三角函數圖形。



SIN 函數的上下限是 $+1$ 到 -1 ，而螢幕由左到右每列可容 64 個字，由上至下可容 16 列，故我們需把 SIN 函數值放大，適當地分配到整個螢幕上面，取螢幕的中央 60 字計，左右各留 2 個空，中央線位於第 32 字處。將函數值放大 30 倍後取其整數加上 32 即得欲印字的位置。

至於間隔的大小要視圖形之精密度決定。若間隔大，譬如每 $\pi/4$ 一個點，則圖形非常粗略。若以 $\pi/2$ 的圖形範圍佈於螢幕上，上下 16 點，故以 $\pi/32$ 為增加值。為了使圖形首尾相銜接，取任何 2π 週期，再造成 loop 即可。

程式

```
10 SELECT R
20 FOR X = -#PI TO #PI STEP #PI/32
30 PRINT TAB(INT(SIN(X)*30)+32); "*"
40 NEXT X
50 END
```

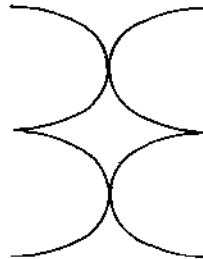
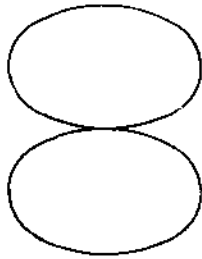
討論

(1)式 30 若改為

```
30 FOR I=1 TO TAB( INT( SIN(X) * 30) + 32): PRINT "*" ; ;
NEXT I: PRINT
```

則可將曲線的左半部填滿 "*" 號。

(2)試做出以下呈對稱的躍動圖形：



以左圖言，其部份程式如下：

```
20 FOR X=0 TO #PI STEP #PI/32
25 Y = INT(SIN(X)*30)
30 PRINT TAB(31-Y); "*" ; TAB(31+Y); "*"
40 NEXT X
```

範例三：固定的資料

題目

由鍵盤輸入一日期，請將月份改以英文名稱印出。

涵意

有時候我們遇到一系列的固定資料，它們可以先設定在程式中，不必等到 RUN 的時候才輸入。例如，行庫的各種存放款利率，水電費率、所得稅率、員工薪資、以及其他等等。處理起來雖然繁，倒也不難，何況我們更可以化繁為簡。

思考

對於固定的一系列資料，除非它們具有規律性，可以用簡單的判斷、運算得到，否則處理的辦法是把它們一系列擺開備用。

把每個月份的名稱依次序擺在 DATA 敘述內。取用的方法有二：一是用 RESTORE 再 READ 到一個變數裏，二是用 FOR loop READ 到一個變數裏，本題，每 RUN 一次只須一個資料，適合前者，若是在 RUN 之中，常常用到各種月份資料，爲了節省時間，適合採用後者。

程式

```
10 DIM A$ 10
20 DATA " JANUARY" , " FEBRUARY" , " MARCH" , " APRIL" ,
    "MAY" , " JUNE" , " JULY" , " AUGUST" , " SEPTEMBER" ,
    "OCTOBER" , " NOVEMBER" , " DECEMBER"
30 INPUT " YYYY, MM, DD" , Y, M, D
40 RESTORE M
50 READ A$
60 PRINT A$; " / " ; D; " , " ; Y
70 END
```

討論

(1)雖然A\$的長度爲10，但是附尾的空白並不會印出，所以日期會緊接在月份後

印出。

(2)假使日期只是一位數，若想以二位數印出，如「5」印成「05」。則須將D用 CONVERT 換成二位數的文字變數D\$。

```
15 DIM D$2
```

```
55 CONVERT D TO D$, (##)
```

```
60 PRINT A$; "/"; D$; ", " ; Y
```

(3)讀者可試一試下列題目：將「5+6=11」在報表上印出成爲「FIVE AND SIX EQUALS ELEVEN」。數目可以任意選取，但加數、被加數及和數皆不超過20。



The Model 2226 CRT/Keyboard Console

範例四：圖案之印製

題目

設欲印製圖案如下：

```
      *
     * * *
    * * * * *
   * * * * * * *
  * * * * * * * *
 * * * * * * * * *
* * * * * * * * * *
 * * * * * * * *
  * * * * * * *
   * * * * *
    * * *
     *
```

函意

設計程式時，我們將問題做系統化的分析，配合程式語言的特性，就能很愉快地完成任務。

思考

這是個有升有降的數列問題，應用 FOR loop 來完成。重點有二：

(1)每列上第一個星號的位置。若計算一下每列上第一個星號的位置，可得

6 5 4 3 2 1 2 3 4 5 6

每數減 1，得

5 4 3 2 1 0 1 2 3 4 5

因此產生此數列的方法為用一個 FOR loop，由 -5 變到 5，然後取其絕對值即可。

(2)每列上星號的個數。由上往下數，星號的個數為

1 3 5 7 9 11 9 7 5 3 1

印表機每次能印一列。故我們必須同時決定印某一系列時的第一個星號位置及星號的個數。而位置的數列已經決定如何產生，所以個數要透過一函數關係產生。觀察第一列的位置是 5，個數是 1；而位置是 1 時個數是 9；位置是 0 時個

數為 11。可看出位置的 2 倍加個數等於 11。由此，用 11 減去位置的 2 倍，即可得個數。

程式

```

10  FOR X=-5 TO 5
20  PRINT TAB(ABS(X));
30  FOR A=1 TO 11-2*ABS(X)
40  PRINT " * ";
50  NEXT A
60  PRINT
70  NEXT X
80  END

```

討論

(1) 根據程式，圖案位於印表機之最左，若欲向右移可更改式 20。向右移 10 字：

```
20 PRINT TAB(ABS(X)+10);
```

(2) 觀察下列二圖案：

```

  * * * *
    * * * *
      * * * *
        * * * *
          * * * *
            * * * *
              * * * *
                * * * *
                  * * * *
                    * * * *
                      * * * *
                        * * * *
                          * * * *
                            * * * *
                              * * * *
                                * * * *
                                  * * * *
                                    * * * *
                                      * * * *
                                        * * * *
                                          * * * *
                                            * * * *
                                              * * * *
                                                * * * *
                                                  * * * *
                                                    * * * *
                                                      * * * *
                                                        * * * *
                                                          * * * *
                                                            * * * *
                                                              * * * *
                                                                * * * *
                                                                  * * * *
                                                                    * * * *
                                                                      * * * *
                                                                        * * * *
                                                                          * * * *
                                                                            * * * *
                                                                              * * * *
                                                                                * * * *
                                                                                  * * * *
                                                                                    * * * *
                                                                                      * * * *
                                                                                      *
                                                                                      * *
                                                                                      * * *
                                                                                      * * * *
                                                                                      * * * *
                                                                                      * * * *
                                                                                      * *
                                                                                      * *
                                                                                      *

```

它們具有相同的數列特性，試以程式設計之。

範例四之二：圖案之印製

題目

設欲印製圖案如下：

```
+++++      +++++      +++++
+++++      +++++      +++++
+++++      +++++      +++++
+++++      +++++      +++++
      +++++      +++++      +++++
      +++++      +++++      +++++
      +++++      +++++      +++++
+++++      +++++      +++++
+++++      +++++      +++++
+++++      +++++      +++++
      +++++      +++++      +++++
      +++++      +++++      +++++
      +++++      +++++      +++++
```

涵意

同範例四之一。

思考

關於此圖案之程式設計問題，依其所安排之位置，分析出一個規則，將整個圖形視為一個 4×6 之矩陣，利用其單數、偶數之別，定其圖案（例單排逢單數，雙排逢偶數圖形為“+”）。待圖案確定後，利用 FOR...NEXT 之 LOOP，連續印 4 次，即得一小長方條圖案。

程式

```
10 REM PLOT THE PICTURE
20 DIM A$(4,6)4,X$4
30 X$="      ": INIT (20)A$( )
40 FOR I=1 TO 4
50 FOR J=1 TO 6
60 GOSUB 140
70 A$(I,J)=X$
80 NEXT J
90 SELECT PRINT 215(144)
100 FOR L=1 TO 4: FOR J=1 TO 6
110 PRINT USING 115,A$(I,J): NEXT J: PRINT HEX(0D0D0A0A): NEXT L
115 %*****
120 NEXT I
130 END
140 I1=I-INT(I/2)*2: I2=J-INT(J/2)*2
150 IF I1(<>0) AND I2(<>0) THEN 180
160 IF I1=0 AND I2=0 THEN 180
170 X$="      ": GOTO 190
180 X$="++++"
190 RETURN
```