

PASCAL
PASCAL 程式設計
PASCAL
PASCAL
PASCAL
PASCAL

李桔亮 編著

力新出版社

PASCAL
PASCAL 程式設計
PASCAL
PASCAL
PASCAL
PASCAL

李桔亮 編著

力 新 出 版 社

PASCAL程式設計

編 者：李 桔 亮

出 版 者：力 新 出 版 社

發 行 者：九 龍 新 山 道 四 三 八 號

承 印 者：力 行 印 刷 公 司

香港柴灣保興工業大廈 8樓 A座

定價：港幣

序

電腦語言是人與電腦溝通的語言，其好壞足以影響人的學習興趣、信心及撰寫程式的效率。一種好的語言應具備易學、易用、易讀的特性，PASCAL 就是朝此目標而設計的，因此在短短的時間裏就能被接受及流行起來。

本書是為欲學習及使用 PASCAL 的人而寫，其編寫目標是：

- (1). 由淺入深循序漸進，使不但適合對其他語言有基礎的人士，也適合初學者及當教本使用。
- (2). 講解力求口語化、層次化，希望讀者很容易了解書中內容，並且讀完後有豁然貫通之感。
- (3). 對於 PASCAL 的設計原理加以詳細的說明，使讀者不僅知其所以也知其所以然。
- (4). 例子多，以便讀者從中學到程式的設計技巧與法則。

在本書編寫期間，承蒙林貴雲細心打字、張銀瓶、林惠芳的校對及程式鍵入，周見運、陳茂生、陳昌蘭、高振銘、王筱青、郭廷明、簡琇瑜等的幫助，才得以讓本書付梓，在此一併誌謝。

本書之編寫及校對雖力求謹慎小心，但恐還有疏漏之處，讀者如於發現錯誤或有其他寶貴意見時，能賜予指教，以便將來修正之用，則不勝感激。

李 桔 亮 謹識

目 錄

第一章 電腦基本概念	1
1 - 1 電腦的基本組織.....	1
1 - 2 資料的儲存方法.....	3
1 - 3 數系之轉換.....	10
1 - 4 程式與資料間的關係.....	11
習 題	12
 第二章 PASCAL簡介	 13
2 - 1 PASCAL發展史.....	13
2 - 2 PASCAL的特性.....	14
2 - 3 PASCAL的格式 (Format)	18
2 - 4 語法結構圖 (Syntax Diagram)	19
2 - 5 PASCAL的組成	22
2 - 6 常變數識別字與 literal	24
習 題	26
 第三章 PASCAL的結構與常變數宣稱(Declaration)	 27
3 - 1 程式的定義.....	27
3 - 2 常數宣稱.....	31
3 - 3 變數宣稱.....	34
習 題	37

第四章 標準的資料型態(Data Type)、系統函數與式子 (Expressions) .. 39

4 - 1	整數型態 (The Type Integer)	39
4 - 1 - 1	整數的運算子 (Operators)	39
4 - 2	實數型態 (The Type Real)	40
4 - 2 - 1	實數的運算子	41
4 - 3	布林型態 (The Type Boolean)	42
4 - 3 - 1	布林型態的運算子	42
4 - 4	文字型態 (The Type Char)	44
4 - 4 - 1	文字型態的運算子	45
4 - 5	函數 (Functions)	45
4 - 6	式子 (Expressions)	48

習 題	52
-----------	----

第五章 簡單的輸出輸入

5 - 1	數字輸入——程序程式 read	55
5 - 2	數字輸入——程序程式 readln	58
5 - 3	文字輸入—— read 和 readln	59
5 - 4	數字與文字混合輸入	60
5 - 5	輸出資料——程序程式 write	62
5 - 5 - 1	整數型態	62
5 - 5 - 2	實數型態	63
5 - 5 - 3	文字型態	65
5 - 5 - 4	布林型態	65
5 - 6	輸出資料—— writeln	66

習 題	69
-----------	----

第六章 指令(Statements)	71
6 - 1 指定指令 (Assignment Statement)	71
6 - 2 複合指令 (Compound Statement)	74
6 - 3 IF 指令	75
6 - 4 CASE 指令	88
6 - 5 REPEAT 指令	90
6 - 6 WHILE 指令	97
6 - 7 FOR 指令	104
6 - 8 程式的佈置 (Layout)	110
習 題	121

第七章 副程式—程序程式與函數程式(Procedures and Functions)	125
7 - 1 程序程式的撰寫及使用	123
7 - 1 - 1 利用變數傳遞資料	126
7 - 1 - 2 利用參數 (Parameters) 傳遞資料	130
7 - 1 - 3 變數與參數 (Variable and Parameter)	136
7 - 1 - 4 程序程式語法 (Syntax)	140
7 - 1 - 5 前向參考 (Forward Reference)	142
7 - 1 - 6 巢狀式的程式塊 (Nested Block)	143
7 - 1 - 7 外在程序程式 (External Procedures)	145
7 - 2 函數程式 (Functions)	145
7 - 3 遞迴性 (Recursive)	149
7 - 4 非當地變數與副作用 (Non-Local Variable and Side-Effects)	169
習 題	171

第八章	純量(Scalars)、子區間(Subranges)與集合(Sets).....	175
8 - 1	純量 (Scalars)	175
8 - 2	子區間 (Subranges)	180
8 - 3	集合 (Sets)	183
習 題		199
第九章	結構性型態(Structured Types).....	203
9 - 1	陣列 (Arrays)	203
9 - 1 - 1	多維陣列 (Multidimensional Arrays)	212
9 - 1 - 2	包裝型陣列 (Packed Arrays)	218
9 - 1 - 3	字串 (Strings)	223
9 - 1 - 4	陣列之語法結構圖.....	225
9 - 2	資料錄型態 (Record Types)	226
9 - 2 - 1	陣列與資料錄混合使用	229
9 - 2 - 2	WITH 指令.....	230
9 - 2 - 3	不定長的資料錄 (Record with Variants)	231
9 - 2 - 4	資料錄的語法結構圖.....	237
習 題		250
第十章	檔案(Files).....	253
10-1	循序檔 (Sequential Files)	254
10-1-1	寫檔案 (Writing to a file)	255
10-1-2	讀檔案 (Reading from a file)	256
10-2	本文檔 (Text Files)	260
10-2-1	自動轉換.....	263

習 題	284
-----------	-----

第十一章 動態的資料結構(Dynamic Data Structures) ... 289

11-1 指標 (Pointers)	288
11-2 鏈串列 (Linked Lists)	292
11-2-1 堆疊 (Stack)	293
11-2-2 貯列 (Queues)	298
11-2-3 一般的加入與刪除	302
11-2-4 串列的遞迴式處理	306
11-2-5 雙向鏈結環 (Double Linked Rings)	307
11-2-6 指標的使用	311
11-3 樹狀結構 (Trees)	312
11-3-1 二元樹 (Binary Trees)	313
11-3-2 一般樹 (General Trees)	317

習 題	335
-----------	-----

第十二章 GOTO指令 339

第十三章 程式範例 345

附錄 A 美國國家標準碼(ASCII)與EBCDIC碼 357

附錄 B PASCAL 語言的字彙(The Vocabulary of PASCAL) 359

附錄 C 語法結構圖(Syntax Diagram) 363

附錄 D 錯誤訊息(Error Message)..... 373

附錄 E PASCAL之參考書籍..... 387

電腦基本概念

1-1 電腦的基本組織

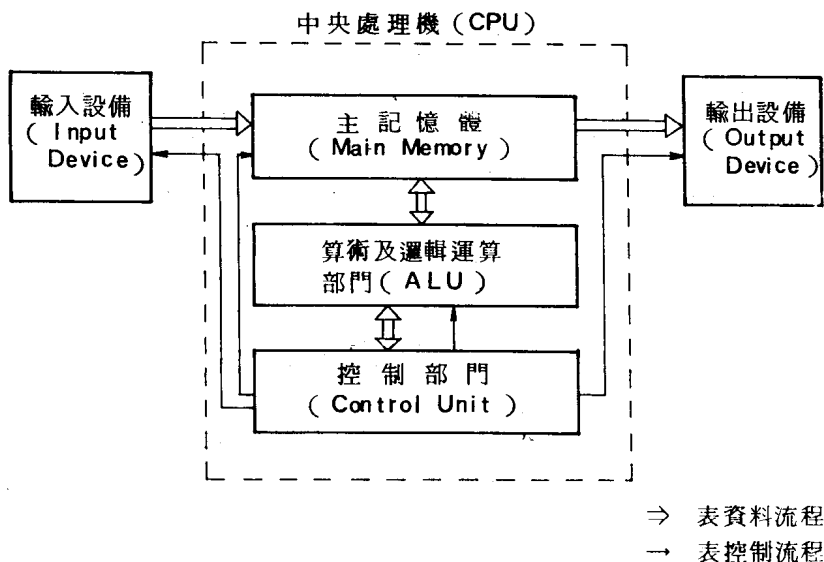


圖 1 - 1 電腦的組織

電腦由五大部門所組成：

- (一) . 輸入設備
- (二) . 輸出設備
- (三) . 主記憶體
- (四) . 算術及邏輯運算部門 (Arithmetic and Logic Unit , 簡稱 ALU)
- (五) . 控制部門

其中，(三)、(四)、(五)項合稱中央處理機 (Central Processing Unit , 簡稱

CPU)。

輸入設備

輸入設備可比喻為人的眼睛、耳朵……等器官，它能將外界的資訊 (Information) 傳送至主記憶體中，以待電腦處理。常用的輸入設備有讀卡機 (Card Reader)、鍵盤 (Keyboard)、磁碟機 (Disk) 及磁帶機 (Magnetic Tape) 等。

輸出設備

輸出設備可比喻為人的嘴巴、手等器官，它能將主記憶體中的資訊傳送至外界以供使用者了解處理結果，或將之儲存於磁碟、磁帶中以供日後再使用。常用的輸出設備有：列表機 (Line Printer)、螢光幕 (Cathode Ray Tube 簡稱 CRT)、磁碟機及磁帶機等。

主記憶體

主記憶體相當於人的腦細胞，它有儲存資訊的功能，不過目前的主記憶體大都用積體電路 (Integrated Circuit) 製造，關機時其中的資訊將消失，不像人的腦細胞有永久儲存的能力。任何待處理的程式或資料均應利用輸入設備輸入並儲存於主記憶體中才可被處理。

主記憶體儲存資料的單位有 Bit、Byte 與 Word 之分：

(一) . Bit

電腦中儲存資料的最小單位，一個 bit 能儲存 0 或 1 兩個數，此兩數是二進位的符號，因此電腦中使用二進位數字系統 (人類日常用 0, 1, 2 ... 9 計數，因此是十進位數字系統)。

(二) . Byte

一個 Byte 由八個 Bits 所組成，是儲存一個文字的單位，如 'A'，'B'，'C'，... '*'，'/'，'+' 等均為文字。

(三) . Word

Word 是主記憶體與算術邏輯運算部門間資料傳送的單位，一個 word

由數個 bytes 所組成，其個數隨著電腦廠牌而不同，如 IBM 370 系列由四個 bytes 所組成，PDP-11 系列由二個 bytes 所組成。

算術及邏輯運算部門

此部門負責做加、減、乘、除……等運算或真、假之邏輯判斷，相當於人的大腦。

控制部門

此部門控制整個電腦的動作，它命令輸入設備讀入資料、算術及邏輯部門做運算、輸出設備輸出資料等。

1-2 資料的儲存方法

主記憶體中有許多位置可用來儲存資料，平常大家所說的電腦有幾 K bytes (1 K = 1024)，幾 K words 指的就是此處位置的大小。爲了在這些位置上正確的拿到所要的資料，主記憶體的位置均有位址編號（大部份電腦以 byte 爲編號單位）、住址編號以二進位數表示，如以 4 個 bits 爲例，其住址編號如下：

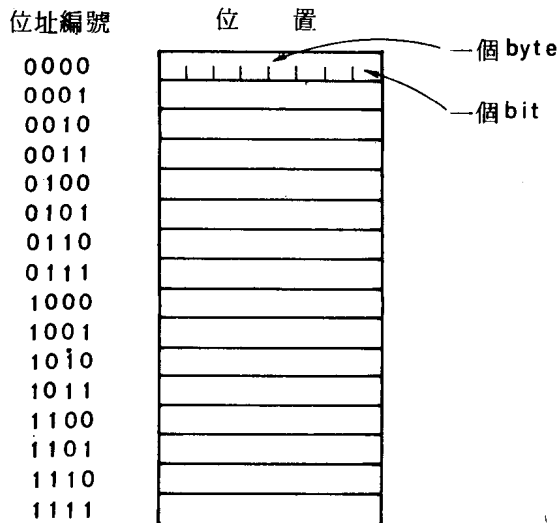


圖 1 - 2 主記憶體中的住址編號

不過這種二進位的住址編號對程式師而言相當不方便，所以在程式設計上大都使用像 A、B、C 等名稱來表示位置，而由電腦自動給予各名稱需要的位置及其住址編號。例如：

```
A = 123
B = 123.45
C = 'D'
```

意思是：命令電腦在主記憶體中找位置，並將其命名為 A，而且把整數 123 存於其中；找位置並將其命名為 B，而且把實數 123.45 存於其中；找位置並將其命名為 C，而且把文字 D 存於其中。

我們知道記憶體中每個 bit 只能儲存 0 與 1 兩個數字，那麼電腦如何儲存像 123、123.45、'D' 等不為 0 與 1 的東西呢？以下分整數、實數及文字的儲存來加以說明。

整數的儲存

爲了方便起見，此處只用 4 個 bits 來解釋整數的儲存，讀者可以依此推演爲 16 個 bits，32 個 bits 或 64 個 bits……等真正情況。

4 個 bits 的所有組合共有 $2^4 = 16$ 種，它們分別是：

```
0000
0001
0010
0011
0100
0101
0110
0111
1000
1001
1010
1011
1100
1101
1110
1111
```

此 16 種組合可以用來代表 16 個數字，當然這 16 個數字中應有 0、正數與負數，否則不能處理三種數中任何一種的電腦有何用處？可是問題在用那些組合來代表正數？那些代表負數？才可以達到最容易運算的效果？電腦中目前使用的方法是：最左邊的 bit（稱為 sign bit）為 0 的組合用來代表 0 或正數，為 1 的組合代表負數。代表正數的組合以二進位的方式計算其所代表的 10 進位數，所以以下的各組合所代表的 10 進位數分別是：

2^3	2^2	2^1	2^0	代表 10 進位數
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7

至於代表負數的每一組合各對應於那一負數呢？電腦中採用的是：某數之 2 的補數（2's Complement）用以代表該數之負數。

何謂 2 的補數？我們用下例來說明：

$$\begin{array}{r}
 1001 \\
 + \quad ? \\
 \hline
 10000
 \end{array}$$

在上例中 1001 加？後會使和變成 10000，此加的數與 1001 稱為互為 2 的補數，也就是相加後其和為 10000 的兩數稱為互為 2 的補數（如果以 8 個 bit 代表整數，就應該是和為 100000000 的兩數稱為互為 2 的補數，讀者可依此類推）。

因為 1 的補數（1's Complement）比 2 的補數好求，所以平常我們均利用 1 的補數來求 2 的補數，底下再以例子來說明何謂 1 的補數。

$$\begin{array}{r}
 1001 \\
 + \quad ? \\
 \hline
 1111
 \end{array}$$

在上例中 1001 加 ? 後會使和變成 1111，此加的數與 1001 稱為互為 1 的補數，也即相加後其和的 4 個 bit 全為 1 的兩數稱為互為 1 的補數。分析上例可知：只要將某數各個 bit 1 變成 0，0 變成 1 即可求得該數之 1 的補數。

研究某數 2 的補數與 1 的補數之關係後可以知道：2 的補數比 1 的補數多 1（因 10000 比 1111 多 1）因此當我們要求某數之 2 的補數時只要先求 1 的補數後再加 1 即可。例如：

1001 之 2 的補數求法如下：

$$\begin{array}{r}
 1001 \\
 (1). \text{先求 1 的補數} \quad 0110 \\
 (2). \text{加 1} \quad \quad \quad + \quad 1 \\
 \hline
 0111
 \end{array}$$

所以 0111 為 1001 之 2 的補數。

電腦中用 2 的補數來代表某數之負數：

所以

-1 用 1 之 2 的補數來代表，其求法如下：

$$\begin{array}{r}
 0001 \\
 (1). \text{求 1 的補數} \quad 1110 \\
 (2). \text{加 1} \quad \quad \quad + \quad 1 \\
 \hline
 1111
 \end{array}$$

亦即用 1111 代表 -1。

-2 用 2 之 2 的補數來代表：

(1). 求 1 的補數

0010

1101

(2). 加 1

$$\begin{array}{r}
 + \quad 1 \\
 \hline
 1110
 \end{array}$$

亦即用 1110 代表 -2。

用同樣的方法可以求出每一負數的表示法如下：

1000	-8
1001	-7
1010	-6
1011	-5
1100	-4
1101	-3
1110	-2
1111	-1

綜合以上的討論，我們將 16 個組合所代表的 16 個數字列表於下：

表 1 - 1 用 4 個 bit 儲存之整數

4 個 bit 之組合	所代表之整數
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	-8
1001	-7
1010	-6
1011	-5
1100	-4
1101	-3
1110	-2
1111	-1