

APPLE II

实用程序解析与应用



北方电脑公司资料组

APPLE II微電腦

**實用程式
解析與應用**

APPLE II 電腦

實用與遊戲 左野用實

目 錄

APPLE A.I.M. — 郵件。檔案之處理	5
APPLE A.R.C. — 多功能資料檔案管理	21
E.M.C. — 電子資料中心	58
迷你打字程式—Tiny Typer	72
使APPLE具有大／小寫顯示功能	81
家庭財務管理程式—H.M.M.	90
矩陣會計—MATRIX ACCOUNTING	114
APPLE磁碟醫生—DISK DOCTOR.....	127
APPLE — TRAP 偵錯程式	134
用電腦學數學	141
APPLE電腦繪圖	164
LOCK SMITH 4.1 正確用法	190
VISICALC的使用方法	204
Visi Dex的使用方法	215
以APPLE担任定時器	223
圖書管理系統程式LIBRARY SYSTEM	228

郵件・檔案之處理

謀傑

前言

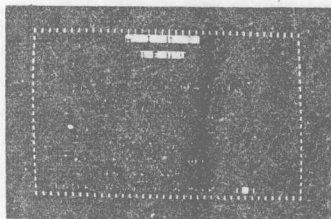
本文所要介紹的 A.I.M. 程式，除了能處理許多你想要要的資料，還可將你親友的姓名、住址和電話號碼保存起來，而且你也可以根據自己的需要而建立屬於自己的 A.I.M.S. (Automated Intelligent Mailing System) 程式。然後藉著程式的執行，你可設定一些程式行，用來儲存公司名稱、住址。此外，還可依據特別的需要而選定一些程式行來作為程式的儲存。

A.I.M. 是用在 48K APPLE II 的 APPLESOFT 上執行。而且需要至少一台磁碟機和一台印表機的配合。

A.I.M. 系統主要是由好幾個獨立的程式模組 (module) 所組成的。因此你會發現到，當你要執行某一個程式模組時，它會將各個 file 都 CATALOG 出來。然後選定一個 file，以執行程式模組。雖然每次要執行時可能皆需輸入相同的 filename。但是如此卻可簡化程式的撰寫。另外，在程式中有一些經常用到的程式行也被安排成副程式 (subroutine)，而被放在程式的前頭以減少程式執行的時間。最後，你將會發現在程式中 get 指令用的相當多。這在許多情況下可以減少我們按 RETURN 鍵的麻煩。

當 A.I.M. 程式一執行的時候，在螢幕上會出現一個 title page，並且有一個指令

Wait-16384, 128 會在跳到 MENU 之前將程式中斷，且等候你輸入任何一個鍵，然後才繼續程式之執行。



定義你要記錄的檔案名稱

MENU 中的第一個程式模組 (initialize new file) 是第一次執行程式時必須選擇的。這個程式模組能容許你在一筆邏輯記錄中，定義你自己所需要的檔案名稱。現在我們就以下圖來說明這種配合的情形。

FIELD#	EXAMPLE 1	EXAMPLE 2	EXAMPLE 3
1	Complete name	Last name	Company name
2	Address	First name	Title
3	City	Middle initial	Complete name
4	State	Phone	Title
5	Zip	Address	Address
6	Phone	City	Address
7	Comment	State	City
8	—	Zip	State
9	—	Phone	Zip
10	—	—	Phone
11	—	—	Comment
12	—	—	Code

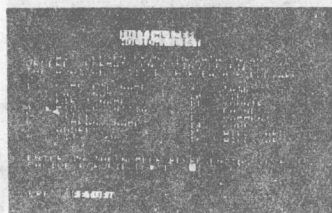
你可能已經注意到，在第三個例子（EXAMPLE 3）中，有二個項目（title 和 address）被重複定義了。但這種情形是容許的，而且在許多的情況下我們也希望如此做。唯一的限制呢？是你必需記住在哪裡發生的項目中，那一個項目是屬於何種性質的。就好像說第一個“title”指的是“Dr.”，或是“Professor”，也可能是“Miss”。然而第二個“title”可能指的是“VP of marketing”，“director of laboratories”，或“sales manager”。

現在我們就上圖所示的第三個例子來看看可以儲存何種資料到邏輯記錄中去，儲存情形如下圖所示。

Company name	NIBBLE MAGAZINE
Title	MRS.
Complete name	ALICE E. NEUMAN
Title	SALES MANAGER
Address	NIBBLE BUILDING
Address	BOX 325
City	LINCOLN
State	MA
Zip	01773
Phone	617-259-9710
Comment	10 SEPT - LAST PAY RAISE
Code	F/C/29+

現在整個 file 都已建立好了，但當你寄出一封信時，信封上欲打出的是 Miss Alice E. Brown，而不是 Sales manager Alice E. Brown 時，要如何處理呢？當然兩者也可都打出，這個問題我們將在後面再加以詳細說明。

雖然這個程式你可以在一筆記錄中可選定 32 個項目。但是我們建議你最好不要超過 14 個項目。因為在 APPLE 電腦中螢幕上祇能容許有 24 行一次同時出現。同時每一個項目皆要佔用磁碟的空間，而且太多的項目會減少了一張磁片所能儲存的記錄筆數。



二個“Wild-card”項目可以容許你做些特殊的需要。當你選擇這二項其中的一個時。此程式會要求你給予此項目一個名字。一些可能的情形如“生日”，“年紀”，“前次拜訪日期”，“信用額度”等。這些都是很實用的，在管理上有相當的意義。

當你完成了在資料庫中對記錄的選定以後，此程式會問你是否滿意這些項目。如果不滿意的話你可以回到原來選定項目的地方，重新選定。如果你批准了這些項目就是你所要的。此程式會要求你對每一個你所選定的項目輸入它應有的長度。其長度應適當即可，太長則會浪費儲存空間。因為我們是以隨機檔（RANDOM file）來儲存的，所以每筆記錄的長度都是固定的，它不像順序檔（Sequential file）一般，不必理會每個項目的長度。因為 Sequential 是變動長度。舉例而言：郵遞區號（ZIP）可以三個字符的長度即可。電話號碼（PHONE）可以 9 個字符即可。

在你對每個項目完成長度的選擇後，你可對所選定的項目和長度重新檢驗一次。假如你對每一件都很滿意，則這些你所選定的資料便儲存在磁片中，並且對資料檔作一個初始化（Initialize）工作（即設你存在資料檔中的資料筆數為 0）。如果你下一個 CATALOG 指令時，你將在螢幕上看到二個似乎是相同的 text file。祇是一個被鎖住了。被鎖住的那個檔所放的就是一個記錄有多少項目（field），這個記錄是多長（Record length），及每個項目的名稱和所對應的長度。而資料檔的第 0 個記錄存放截至目前為止，有多少的記錄

已經存在此資料檔中。而且被上鎖的檔，其檔名多了CTR-B。因此要殺掉這個檔必需輸入：
UNLOCK ABC CTR-B CR 假設檔名
DELETE ABC CTR-B CR 稱叫ABC
CR(Carriage Return)即按RETURN鍵
到此為止你已經將所需要的資料項目定義好，接下來就可以開始輸入客戶資料了。

輸入要儲存之檔案資料名稱

從MENU中選擇第二個模組(ENTER DATA)。然後輸入你資料檔所在的slot和Driver NO.。並且程式會自動Catalog所有的檔案，以供你選擇。輸入你所需的檔名且按CR (即按RETURN鍵)。假如你祇按CR而無任何的檔名輸入。則MENU會自動出現在螢幕上，如此做是方便於讓你曉得你輸入了錯誤的檔名。如果沒有錯誤的檔名，則螢幕上出現了第一個項目名稱且帶有一個有一定距離的中括弧。這個距離等於你所選定此項目的長度。因此你對此項目所輸入的資料長度必須在這個範圍之內。如果你輸入的資料超過了右括弧則會被警告說，你的資料太長了，重新輸入一次。對第一個項目而言，按RETURN則會打斷這個模組的執行，MENU會出現在螢幕上。而小於其他項目而言若按RETURN則是一種合理的輸入方式，祇是意味着對這個項目不給予它任何的資料。

當這筆記錄就是你所要的，則存到磁片並且指派一個記錄編號(Record NO.)給予此記錄，如果你用第四個程式模組"Change/delete DATA"則可以改變記錄編號，此點後面有詳細說明。

假如你在未按RETURN之前發現有錯誤存在，祇要按退後鍵(BACK SPACE)回來修改即可。如果是在按RETURN之後才發現錯誤，則程式容許你在對每個項目皆輸入完畢之後詢問你是否有要修改的項目，此時輸入那欲修改的那個項目，回去修改此項目即可。

資料顯示

MENU中的第三個程式模組"DISPLAY/PRINT DATA"可供你顯示(在螢幕上)，印出(在印表機上)，目前儲存在磁片中的資料。假如你要將資料印出則必需先輸入你的Printer Card是插在那一個slot上。然後輸入要開始之記錄編號(Record NO.)。每當程式提出一筆記錄後，下一筆記錄可按(→)鍵讀取，或是按(←)鍵讀取上一筆記錄，或許你想跳到中間讀取，這樣也可以，祇要按(ESC)後再輸入你所想要讀取的Record NO。如果按RETURN鍵則會跳到(顯示/印出)這個選擇中，對於此項選擇祇按RETURN則MENU會自動出現在螢幕上。

改變/剔除資料

第四個程式模組(CHANCE/DELETE DATA)容許你更新或改變在磁片中的任何資料。對於某一筆記錄，你覺得已無使用價值時則可以剔除它。要執行這個程式模組之前，你必需先明白是那一個record NO.有問題。(在你把資料放到磁片中時都會有一個編號)。我們可以利用第三個程式模組印出所有的資料和編號，以供檢查有無錯誤。有了record NO程式就可以將它提取出來以供剔除或者是修改。假如你要剔除此筆記錄，則需照著螢幕上所指示的回答"YES"而不能祇回答一個字"Y"，如果不是回答"YES"則此筆記錄不會被刪除，如此一來即可防止失誤，而導致資料的喪失。然而剔除是讀取此資料檔的最後一筆記錄放到欲剔除的record中。然後將record NO為0的值減一，再儲存到record NO.為0中，我們舉例說明如下：假設此file已有39筆record且#17欲剔除。第一：先到record 0中讀取39出來。第二：將record 39的資料提出來。第三：將提出來的儲存到record 17中。第四：將39-1=38儲存到record 0中。如此完成了剔除的程序，不但是record 17被蓋住了而且不會浪費空間，因為以後新增資料時可從第39個record開始。

假如是改變而不是剔除記錄，則祇要挑選

欲修改的項目加以修改即可，按 RETURN 則會跳開修改程式回到 MENU 中。因為程式中皆用矩陣的方式來儲存此種資料，所以只要給予註標可直接加以修改。

資料之組合

第六個程式模組 (PRINT MAIL LABELS) 是這個程式的核心，同時使用起來也很方便，而且其工作能力也會令你吃驚。利用程式所印出來的報表可以是你所定義的各個項目。假如需要的話，這些項目也可重覆許多次。第一個步驟是輸入你所要印出來之項目的代號。而這些代號就是視你所要印出的型式而定。現在讓我們用第三個例子作為指引且以如下的型式印出報表。

Mrs. Alice E. Neuman
Sales Manager
Nibble Magazine
Box 325
Lincoln, MA 01773

這個報表是由如下的項目和選定順序來印製產生的。

ORDER OF CHOICE	FIELD #	NAME OF FIELD
1	2	Title
2	3	Complete name
3	4	Title
4	1	Company name
5	5	Address
6	7	City
7	8	State
8	9	Zip

項目的選擇已經決定，接下來便是是否要將兩個或多個項目併在同一行中印出來。在這個例子中我們首先將 title 和 complete name 併成了 "Mrs. Alice E. Brown"。除此之外，還將 City 到 state, state 到 ZIP 拼成了印表的最後一行。程式會問你是那幾行要拼在一起。記住：祇有連續的行能合併。所以我們祇能選行(1)與行(2)，而不能選行(1)與行(3)。所以你必需花費一點時間來決定選用的順序。行(1)與行(2)的連接是一個 space。所以你

祇要按一個空間棒再按 RETURN 即可。按 RETURN 的意思即為你已完成了行(1)與行(2)之間的連接。如果祇按 RETURN 則會打斷項目之間的連接。報表的最後一行的行(8)與行(7)之間一個逗點和一個空白。行(7)與行(8)之間是二個空白。當你完成了行與行之間的連接時則按 RETURN，這也是電腦唯一能知道你已完成了連接工作的途徑。當你完成對報表格式的定義後，一個樣本會出現在螢幕上並且等待你的檢查。

最後第七個程式模組 (END PROGRAM) 會停止程式的執行並且提醒你對你所輸入的資料別忘了做一份拷貝。

將 A.I.M. 程式改成別的作用

將 A.I.M. 修改是非常容易的一件事情，雖然它是一個 mail-list 的程式，但是只要變更程式中的幾行便可成為你的個人圖書資料管理和庫存系統。欲將 A.I.M. 系統變更為圖書管理系統，祇需更改如下：

```
580 DATA "PRINT REPORT", "END
PROGRAM"
1010 V = 10: REM NO. OF FIELD ITEMS
1040 DATA "BOOK TITLE", "AUTHOR",
"PUBLISHER", "DATE", "SUBJECT",
"CATEGORY", "COMMENT", "MISC",
"WILDCARD", "WILDCARD"
6010 A1$ = "PRINT BOOK LIST":
B1$ = "REPORT"
```

若要作為個人庫存系統，則僅需變更如下
的幾行便可達成：

```
580 DATA "PRINT REPORT", "END
PROGRAM"
1010 V = 10: REM NO. OF FIELD ITEMS
1040 DATA "ITEM", "PURCHASE DATE",
"COST", "PLACE OF PUR-
CHASE", "SERIAL #",
"DEPRECIATION %", "CODE",
"MISC", "WILDCARD",
"WILDCARD"
6010 A1$ = "PRINT INVENTORY LIST":
B1$ = "REPORT"
```

THE INS AND OUTS OF A.I.M.

前面已將整個 A.I.M. 程式的功能及用法，作了大略的介紹。下面就來說明 A.I.M 程式中各程式段所擔任的工作。

輸入

程式行 (line) 100 設定兩個記憶位置的初值，而這兩個記憶位置是用來儲存你所使用的磁碟機 (drive) 和週邊連接器 (slot) 的號碼。行 120~150 則會將本文所介紹程式的一段前導文，置於螢幕上。然後程式會跳到行 500 去，以開始對程式中的程各個常數下定義。例如在行 530 我們定義 BC\$=CTRL-B (被用來附加到檔案名稱上去)，GS=CTRL-G (嗶聲)，HS=CTRL-H (退後鍵)，V1\$=(左括弧)，且 V2\$=(右括弧)。

行 1000 開始副程式的初值化，在行 1010 的變數 V 表示在程式中所使用項目頭銜 (file title) 的數目，這些項目頭銜被置於行 1040。假如你要改變或增加這個程式行，一定要確信所使用的項目頭銜是一個偶數。行 1090 是要求一個偶數 V。

行 1070 是一個有趣的常式，其主要作用是将螢幕顯示集中成一行。而 POKE 指述設定了螢幕的窗口 (window) 範圍，因此在螢幕上方的一些字，不致於因螢幕之上捲 (scroll) 動作而消失。

行 1130 是程式中經常用到的常式。當使用者 (user) 輸入一個值時，程式將會以字串變數 (T\$) 的形式接收，然後將字串 T\$ 轉換成一個數字的形式且和程式所容許的數值相比較。這種技巧可避免將數字打成文字或將文字打成數字所產生的錯誤。行 1140 會接受你指定給 Wild Card 項目的名稱。一旦所有的項目都定義好了之後，行 1200 會將它們在螢幕上顯示出來給你看。次一行的程式則會要求你查對這些項目是否錯誤。注意：行 1210 使用 GET 指令，每當使用 GET 指令時，其後面最好是跟著 PRINT 指述，否則 DOS 將會無法辨別。

行 1230-1320 會要求你輸入各定義項目

的長度，然後才要求你判斷整個項目的記錄結構是否正確。

假如上述記錄結構的選定是正確的，則行 1330-1370 會將這些項目寫入磁碟。這是一種順序檔 (sequential file) 的形式，檔案名稱的尾端都有附加一個 CTRL-B，而一個隨機存取檔 (random access file) 則是從 0 開始指示記錄。現在初值化的工作已經完成，所以程式會再回到 MENU 去。

行 2000 開始資料輸入。隨機檔被打開且檔案中各項目的記憶資料被讀入變數 REC 中，置於行 200-210 的副程式會接收新資料，且將其送到符號為 P\$ 的陣列中。注意：行 2090 會檢查 P\$ 陣列中的第一個陣列 P\$(1) 是否有輸入，假如沒有，則表示資料輸入工作已經完成，而且程式又回到 MENU 去。假如有資料輸入，則行 2100 會查對資料的長度是否會太長。假如不會，則整個陣列都會在螢幕上顯示出來，而且將會先被寫入磁碟中。在這個查對期間，你可以修改任何不正確的程式行或將輸入操作完全放棄。一旦你認可了你輸入的各項目資料，則這些資料會被寫入磁碟，然後程式又回到資料的輸入型式，以接受其它資料的輸入。

行 4000 開始的程式，主要是具有修改／剔除資料的功能。當你告訴電腦你所要修改的那筆記錄名稱，則電腦會由磁碟中將那筆記錄叫出來，且利用行 4050-4130 將那筆記錄在螢幕上顯示出來。此時，你可以修改／剔除這筆資料。行 4200 要求一個 "YES" 的輸入以要求剔除資料。假如輸入 YES，則在檔案中的最後一筆記錄由行 4250 讀入記憶器中，然後由行 4270 再次寫入磁碟中以剔除那一筆記錄。此時整個記錄的總數被減 1，且這個新值也由行 4290 寫入磁碟中。就此點而言，你又回到了修改／剔除資料的輸入選擇狀態。

行 4330 開始的程式，是用來修改記錄的資料。這行程式會要求你輸入要修改的程式行號。行 4350 會接收正確的資料，且行 4360 會檢查輸入資料的長度，以確定此資料不會太

長。假如上述程序沒有問題，則行 4390 會將這些資料寫入磁碟，而且再回到修改／剔除的輸入選擇型態。

書信一標題常式是由行 6000 開始。要印出的項目資料則由行 6060 送到螢幕顯示。在你決定標題中的某一行是你所要的之後，可以利用行 6080 來輸入它們。行 6090 會搜尋一個空白 (null) 的輸入 (即 RETURN 鍵)，它主要是用來表示你已完成了要印出之各項目的選擇工作。行 6130-6140 會將你所選擇的那些項目印出來，以允許你查對一下所印出的是否符合你所選擇的。

行 6180 會問你是否希望將各項目連接起來。假如你決定要，則行 6190 利用陣列 TX\$ 來儲存所有的游標返回 (carriage return)，且這個參數會在每個資料項目之後被印出。

行 6250 要求你輸入欲連接的各項目，此種連接方式會將兩個項目連接在一起。行

6260 則會將每個輸入以 T1\$ 的形式接收且和一個 C/R (CHR\$(13)) 相比較。假如輸入不是一個 C/R (Carriage Return)，則此輸入會和 TX\$ 連接起來，而且若需要的話還可再執行其它的项目連接。

最後的結果是一個陣列元素 TX\$，此陣列元素含有你利用 C/R 來連接兩個輸入所需的所有的字母、數字或特殊的符號。行 6300-6330 使一個標題樣本在螢幕上顯示出來，因

此你可以看到你所設計的書信標題形式。

標題印出常式是置於行 6700-6770。在你標題中的每個資料項目 (P\$) 被印出，而且其後跟隨著連接陣列 (TX\$)。變數 R1 表示在標題中被印出行的實際數目。這個變數 R1 在每次執行 C/R 後，就會被加 1。且當標題的印出工作完成後，電腦會使用 R1 來指示現標題和下一個標題中空白間隔所需之游標返回 (Carriage Return) 的數目。在所有的標題印完後，程式又回到 MENU 去。

輸出

首先是儲存你設計完成的標題格式。然後每當你要印出標題時，你必須重新輸入格式。這是現在 A.I.M. 的操作方式。假如你能將格式存入磁碟中，且在需要時再將它叫出來，則將可以節省很多的時間而且非常有效。

資料輸入型態對於一些特定的項目，可以在程式中預先定義。例如，當輸入兩個代表州的字母時，假如你忘了懷俄明州的代碼 (code) 時，壓下 CTRL-A，則一些州的縮寫列表將會在螢幕上顯示出來。這項功能可以應用到一些需要縮寫的項目中。

最後，一個自動撥號 (autodial) 的特性，將會使具有 Modems 的人感到興趣。因為名字可以置於 Find 常式中且自動替你撥號。(在 MENU 第 5 項中)

本文之全部程式，連同磁碟每份 200 元 (如需郵寄者，另加郵費 10 元)，歡迎函購。

APPLE A.I.M.

ALIST

```
1 PRINT " ": PRINT " "
100 POKE 768,0: POKE 769,0
110 REM PLACE TITLE ON SCREEN
120 TEXT: HOME: GOSUB 340
130 VTAB 3: HTAB 10: PRINT "M I C R O - S O F T W A R E"
140 C": VTAB 4: HTAB 14: PRINT "P.O. BOX 325": VTAB
150 5: HTAB 11: PRINT "LINCOLN, NS 01773"
```

768 存 DATA file 所在的 slot NO.

769 存 DATA file 所在的 driver NO.

Subroutine 340: 以星號在 screen 上畫框框。

```

140 VTAB 9: HTAB 13: PRINT "P R E S E N T E": VTAB
12: HTAB 12: FLASH: PRINT "*** A. I. M. S. *
*": NORMAL: VTAB 15: HTAB 10: PRINT "AUTOMA
TED INTELLIGENT": HTAB 13: PRINT "MAILING SY
STEM": VTAB 20: HTAB 7: PRINT "COPYRIGHT 198
0, C.D.S., INC"

150 VTAB 22: INVERSE: HTAB 10: PRINT "PRESS RET
URN FOR MENU": NORMAL

160 WAIT - 16384, 128 ← 此指令為等候 user key in 任何一個鍵
170 GOTO 500

180 REM *** SUBROUTINES BELOW

190 VTAB 23: CALL - 868: PRINT "CORRECT(Y/N)": B
$: GET Y$: PRINT Y$: RETURN

200 PRINT "ENTER ": INVERSE: CV = PEEK (37): GOSUB
210: P$(K) = Q$: RETURN

210 PRINT T$(K): NORMAL: PRINT ": ": G$: PRINT
V1$: TAB (TV(K) + 3): V2$: VTAB (CV + 2): HTAB
2: INPUT Q$: VTAB (CV + 1): HTAB (LEN (T$(K
)) + 9): CALL - 958: PRINT Q$: RETURN

220 PRINT D$: "OPEN ": FILE$: ", L": RL
230 PRINT D$: "WRITE": FILE$: ", R": RX
240 RETURN

250 PRINT "PRESS": INVERSE: PRINT "RETURN": NORMAL
: PRINT "TO QUIT": RETURN

260 VTAB 2: HTAB (20 - LEN (AA$) / 2): INVERSE
: PRINT AA$: NORMAL

270 VTAB 4: HTAB 2: PRINT B$: VTAB 4: INVERSE: HTAB
(20 - LEN (A1$) / 2): PRINT A1$: POKE 34, 4:
NORMAL: RETURN

280 PRINT D$: "OPEN ": FILE$: ", L": RL
290 PRINT D$: "READ ": FILE$: ", R": RX
300 RETURN

310 HOME: PRINT: FOR K = 1 TO NF: PRINT K: ", "
: P$(K): " - ": INVERSE: PRINT P$(K): NORMAL
: NEXT K: RETURN

320 VTAB 23: HTAB 10: GOSUB 250: PRINT: VTAB 21
: PRINT "ENTER LINE # OF INCORRECT DATA: ": B
$: INPUT Z$: IF Z$ = " " THEN 2160

330 HOME: VTAB 7: PRINT T$(Z): " - ": INVERSE:
PRINT P$(Z): NORMAL: PRINT: PRINT "ENTER
IN CORRECT DATA: ": B$: INPUT P$(Z): RETURN

340 VTAB 23: FOR K = 1 TO 39: PRINT "*": NEXT K
: PRINT: VTAB 1: FOR K = 1 TO 39: PRINT "*"
: NEXT K: PRINT

350 FOR K = 1 TO 22: PRINT "*": HTAB 39: PRINT
*: NEXT K: RETURN

360 HOME: PRINT D$: "CATALOG, S": SX: ", D": DX
370 PRINT: PRINT "ENTER FILE NAME: ": B$: INPUT
FILE$: IF LEN (FILE$) = 0 THEN CLEAR: GOTO
500

```

```

380 PRINT D$;"OPEN";FILE$ + BC$
390 PRINT D$;"READ";FILE$ + BC$
400 INPUT NF: INPUT RL: DIM T$(2 * NF),TV(NF),P$
    (NF)
410 FOR K = 1 TO NF: INPUT T$(K): INPUT TV(K): NEXT
    K
420 PRINT D$;"CLOSE";FILE$ + BC$
430 RETURN
440 PRINT D$;"OPEN ";FILE$;" ,L";RL
450 PRINT D$;"READ ";FILE$;" ,R0"
460 RETURN
470 TX$(LX) = " ": REM NOTHING IN BETWEEN " "
480 GET T1$: PRINT T1$;: IF T1$ = CHR$(13) THEN
    D RETURN
490 TX$(LX) = TX$(LX) + T1$: GOTO 480
500 REM
510 REM MENU
520 V1 = 7: DIM M$(V1):B$ = "
    ": REM 30 SPACES
530 BC$ = CHR$(2):D$ = CHR$(4):G$ = CHR$(7)
    :H$ = CHR$(8):V1$ = CHR$(9):V2$ = CHR$(
    93)
540 TEXT : HOME : GOSUB 340
550 AA$ = "A. I. M. S.":A1$ = "M E N U": GOSUB 26
    0: RESTORE
560 FOR K = 1 TO V1: READ M$(K): NEXT K
570 DATA "INITIALIZE NEW FILE","ENTER DATA","DI
    SPLAY/PRINT DATA","CHANGE/DELETE DATA","FIND
    DATA"
580 DATA "PRINT MAIL LABELS","END PROGRAM"
590 FOR K = 1 TO V1: VTAB (2 * K + 6): HTAB 8: PRINT
    K$;" ,M$(K): NEXT K
600 VTAB 22: HTAB 4: PRINT "ENTER CHOICE (BY NUM
    BER)":G$:V1$;: "V2$;H$;H$: NORMAL : GET Y
    $: PRINT Y$:Y = VAL (Y$): IF Y < 1 OR Y > V
    1 THEN 600
610 VTAB 2 * Y + 6: HTAB 8: INVERSE : PRINT Y$;"
    ":M$(Y): NORMAL : FOR KK = 1 TO 400: NEXT KK
620 IF Y = 7 THEN 7000
630 SX = PEEK (768):DX = PEEK (769): IF SX > 0 AND
    SX < 8 AND DX = 1 OR DX = 2 THEN 710
640 TEXT : HOME : VTAB 6: CALL - 958: PRINT "EN
    TER SLOT # FOR DATA DISKETTE":G$;: GET SX$:
    PRINT SX$:SX = VAL (SX$)
650 IF SX < 1 OR SX > 7 THEN 640
660 VTAB 8: CALL - 868: PRINT "ENTER DRIVE # FO
    R DATA DISKETTE":G$;: GET DX$: PRINT DX$:DX
    = VAL (DX$)
670 IF (DX - 1) * (DX - 2) < > 0 THEN 660

```

line 500-610 顯示MENU於screen上，並選擇之。

ARRAY M\$ 用以存放七個標題名稱

BC\$為CTR-B，C\$為Bell，H\$為back space，V1\$為左括弧，V2\$為右括弧。

Subroutine 260 為將AA\$放在VTAB2的中間，並以INVERSE表示。

將DATA file 所在的slot NO. 放在SX driver NO. 放在DX。

CALL-958 為將游標所在以後的下方清除。

line 640 至 660 要求 user 由鍵盤上 keyin slot NO. and driver NO.

line 670 判斷如果 driver NO. 不是 1 或者不是 2 則繼續輸入。

```

680 VTAB 10: CALL - 958: PRINT "SLOT # =";SX: PRINT
    "DRIVE # =";DX: GOSUB 190: IF Y$ = "N" THEN
690 IF Y$ < > "Y" THEN 640
700 POKE 768,SX: POKE 769,DX
710 ON Y GOTO 1000,2000,3000,4000,5000,6000,7000

```

```

1000 REM INITIALIZER SUBROUTINE
1010 V = 16: REM V=NO. OF FIELD TITLES, EVEN NO.

```

ARRAY T\$ 用以存放可供選用來定義的
16個 field name。
ARRAY T 存放經定義的 field NO。
ARRAY TV 存放經定義 field NO. 的長度。

```

1020 DIM T$(V),T(2 * V),TV(2 * V)
1030 FOR K = 1 TO V: READ T$(K):: NEXT K:T$(0) =
    H$ + H$ + H$ + H$ + LEFT$(B$,4)
1040 DATA "COMPLETE NAME","LAST NAME","FIRST NA
    ME","MIDDLE INITIAL","TITLE","COMPANY NAME",
    "ADDRESS","CITY","STATE","ZIP","COUNTRY","PH
    ONE","COMMENT","CODE","WILD CARD","WILD CARD
    "

```

line 1030 將各個 field name
讀入 ARRAY T\$ 中。

```

1050 TEXT : HOME : PRINT "ENTER NAME OF FILE:";G
    $:: INPUT FILE$
1060 HOME : INVERSE : HTAB 14: PRINT "INITIALIZE
    R"
1070 HTAB (20 - LEN (FILE$) / 2): PRINT FILE$: POKE
    34,3: NORMAL
1080 HOME : PRINT "YOU WILL BE ENTERING IN DATA
    IN AN": PRINT "ORDERED FORMAT CONSISTING OF
    FIELDS.": PRINT "THE AVAILABLE TITLES FOR EA
    CH FIELD ARE:"

```

```

1090 FOR K = 1 TO V / 2: PRINT K;". ";T$(K):: HTAB
    (24): PRINT K + V / 2". ";T$(K + V / 2): NEXT
    K: PRINT : PRINT

```

將各個 field name 和所對應的 field NO.
以：

```

1 complete name 9 STATE
2 CAST NAME 10 ZIP
: :

```

的型式顯示出來。

```

1100 N = 0
1110 N = N + 1
1120 VTAB 19: CALL - 958: PRINT "ENTER IN THE N
    UMBER REPRESENTING YOUR CHOICE FOR FIELD #";
    N;G$: VTAB 23: GOSUB 250: VTAB 19: HTAB 22: INPUT
    ": ";T$: IF LEN (T$) = 0 THEN 1160

```

Subroutine 250 為在 VTAB 23 中顯示
Press Return to Quit

```

1130 T = VAL (T$): IF T < 1 OR T > V THEN 1120
1140 IF T = V OR T = V - 1 THEN PRINT "ENTER NA
    ME FOR THIS FIELD:";G$: INPUT Z$:T(N) = T:T
    $(T(N)) = Z$: GOTO 1110

```

line 1140 為如果 user 選定 field NO.
15 或者是 16 則由 user 輸入 field NAME
以取代 wild Card 這個名稱。

```

1150 T(N) = T: GOTO 1110
1160 HOME :N = N - 1
1170 IF INT (N / 2) = N / 2 THEN N2% = N / 2
1180 IF INT (N / 2) < > N / 2 THEN N2% = N / 2
    + 1

```

此段作用為將所選定的 field 數除以在
line 1200 中可做到 line 1090 的功能。

```

1190 FOR K = 1 TO N2%
1200 PRINT K;". ";T$(T(K)):: HTAB (21): PRINT K +
    N2%;". ";T$(T(K + N2%)): NEXT K
1210 VTAB 22: CALL - 868: PRINT "CORRECT(Y/N)";
    G$: GET Y$: PRINT Y$: IF Y$ = "N" THEN 1080

```

```

1220 IF Y$ < > "Y" THEN 1210
1230 HOME : VTAB 4: PRINT "ENTER LENGTH VALUES FOR EACH FIELD:"
1240 FOR K = 1 TO N
1250 VTAB 6: CALL - 95B: PRINT "FOR FIELD # " : K
      " " : ; / INVERSE : PRINT T$(T(K)); G$ : NORMAL
      : INPUT " " : ; TV(K) : NEXT K
1260 TEXT : HOME : INVERSE : HTAB (20 - LEN (A$) / 2): PRINT A$ : PRINT : PRINT : POKE 34, 3:
      NORMAL
1270 RL = 0: PRINT " " FLD NAME FLD LE
      NGTH"
1280 PRINT "-----"
1290 FOR K = 1 TO N
1300 PRINT K: " " : HTAB 5: INVERSE : PRINT T$(T(K)); : NORMAL : HTAB 25: PRINT " " : ; TV(K) : RL =
      RL + TV(K) : NEXT K
1310 VTAB 23: CALL - 86B: PRINT "CORRECT (Y/N)
      " : G$ : GET Y$: PRINT Y$: IF Y$ = "N" THEN 1230
1320 IF Y$ < > "Y" THEN 1310
1330 PRINT D$ : "OPEN" : FILE$ + BC$ : "S" : SX : "D" : DX
1340 PRINT D$ : "WRITE" : FILE$ + BC$
1350 PRINT N: PRINT RL: FOR K = 1 TO N: PRINT T$(T(K)) : PRINT TV(K) : NEXT K
1360 PRINT D$ : "CLOSE" : FILE$ + BC$
1370 PRINT D$ : "LOCK" : FILE$ + BC$
1380 PRINT D$ : "OPEN" : FILE$ : "L" : RL
1390 PRINT D$ : "WRITE " : FILE$ : "R"
1400 PRINT 0
1410 PRINT D$ : "CLOSE"
1420 CLEAR : GOTO 500
2000 REM DAA ENTRY SUBROUTINE
2010 A1$ = "DATA ENTRY ROUTINE"
2020 TEXT : HOME : GOSUB 260: HOME : GOSUB 360: R
      X = 0: GOSUB 280
2030 INPUT REC: REC = REC + 1
2040 PRINT D$ : "CLOSE" : FILE$
2050 HOME : PRINT : PRINT "THIS WILL BE RECORD #
      " : REC
2060 VTAB 23: PRINT "RETURN ON" : ; INVERSE : PRINT
      T$(1) : ; NORMAL : PRINT "TO QUIT" : VTAB 8
2070 FOR K = 1 TO NF
2080 GOSUB 200
2090 IF LEN (P$(1)) = 0 THEN 2200
2100 IF LEN (P$(K)) < = TV(K) THEN 2120
2110 PRINT "ENTRY TOO LONG" : G$ : G$ : FOR KK = 1 TO
      750: NEXT KK: VTAB (CV + 1): HTAB 1: CALL -
      95B: GOTO 2080
2120 NEXT K
2130 GOSUB 190

```

作一個迴路將所選定的各個 field 輸入其 field length。

先將 RL 歸 0，因為 RL 為各個 field length 的總合就是 record length。

line 1290 至 1300 將經選定的 field NO, field name 和 length 再顯示一次供 line 1310 的 user 批准。

line 1330 至 1370 將以你所輸入的 file name Append 一個 CTR-B 來開檔，並將總 field 等數，record length，各個 file name 和 field length save 到 file 中並上鎖。

line 1380 至 line 1410 開一個 random file，並將其 record (記錄已有多少 record save 在 file 中) 歸為零。

Subroutine 360 為開檔，並以 RX 為 record NO，準備讀取 data。

Subroutine 200 為一次顯示出在 initial New file 中被選定為 field name。並且緊接著各個 field name 由 user keyin 的資料。

假如第一個 field 祇按 return 而無任何資料則跳到 line 2200 作再顯示一次你所 keyin 的資料並再檢查一次。

檢查所輸入的各個 field 其長度是否超過在 initial new file name 中所定義的長度。

Subroutine 190 為由 user 判斷是否正確？

```

2140 IF Y$ = "N" THEN 2250
2150 IF Y$ < > "Y" THEN 2130
2160 RX = REC: GOSUB 220

```

Subroutine 220 為開檔並以 RX 為 record
NO. 準備將 data save 到 file.

```

2170 FOR KK = 1 TO NF: PRINT P$(KK): NEXT KK
2180 PRINT D$;"CLOSE"
2190 REC = REC + 1: GOTO 2050

```

```
2200 REC = REC - 1: RX = 0
```

```
2210 GOSUB 220
```

```
2220 PRINT REC
```

```
2230 PRINT D$;"CLOSE"
```

```
2240 CLEAR: GOTO 500
```

```
2250 GOSUB 310: PRINT NF + 1: "ABORT"
```

```
2260 VTAB 23: HTAB 10: GOSUB 250: PRINT: VTAB 2
```

```
1: PRINT "ENTER LINE # OF INCORRECT DATA: ";
```

```
G$: INPUT Z$: Z = VAL (Z$)
```

```
2270 IF Z$ = "" THEN 2160
```

```
2280 IF Z = NF + 1 THEN 2050
```

```
2290 IF Z < 1 OR Z > NF THEN 2250
```

```
2300 GOSUB 330: GOTO 2250
```

```
3000 REM DISPLAY/PRINT DATA
```

```
3010 A1$ = "DISPLAY/PRINT DATA"
```

```
3020 TEXT: HOME: GOSUB 260: HOME: GOSUB 360: R
```

```
X = 0: GOSUB 280
```

```
3030 INPUT REC
```

```
3040 PRINT D$;"CLOSE": FILE$
```

```
3050 HOME: VTAB 6: PRINT "ENTER IN PRINTER INFO
```

```
RMATION.": PRINT: PRINT "DO YOU HAVE A PRIN
```

```
TER (Y/N)": G$: GET Y$: PRINT Y$: IF Y$ = "N
```

```
" THEN SLOT = 0: GOTO 3140
```

```
3060 IF Y$ < > "Y" THEN 3050
```

```
3070 VTAB 10: CALL - 868: PRINT "ENTER IN SLOT
```

```
# ": G$: GET SL$: PRINT SL$: SLOT = VAL (SL$
```

```
)
```

```
3080 IF Y$ = CHR$ (27) OR Y$ = CHR$ (32) THEN
```

```
CLEAR: GOTO 500
```

```
3090 IF SLOT < 1 OR SLOT > 7 THEN PRINT G$: G$: GOTO
```

```
3070
```

```
3100 PRINT: PRINT "PLEASE TURN ON PRINTER.": PRINT
```

```
"PRESS ": INVERSE: PRINT "RETURN": NORMAL
```

```
: PRINT "WHEN READY.": G$: INPUT Y$
```

```
3110 PRINT D$;"PR#": SLOT
```

```
3120 PRINT
```

```
3130 PRINT D$;"PR#0"
```

```
3140 HOME: VTAB 22: HTAB 10: GOSUB 250: VTAB 7:
```

```
PRINT
```

```
3150 PRINT "VIEW(V) OR PRINT (P) DATA ": G$: GET
```

```
VP$: PRINT VP$: IF VP$ = CHR$ (13) THEN CLEAR
```

```
: GOTO 500
```

```
3160 PRINT: CALL - 958: PRINT "ENTER RECORD #
```

```
TO START WITH ": G$: INPUT START$: RX = VAL
```

```
(START$)
```

Subroutine 360 為將在磁片上的 file
name catalog 出來, 以供選擇.

```

3170 IF RX > REC THEN PRINT "ONLY"; REC; "RECORDS
      ON FILE!"; G%G%: GOTO 3160
3180 IF RX < 1 THEN RX = 1
3190 GOSUB 280
3200 FOR K = 1 TO NF: INPUT P%(K): NEXT K
3210 IF VP% = "P" THEN PRINT D%; "PR%"; SLOT: PRINT
      : PRINT "RECORD #"; RX
3220 GOSUB 310
3230 PRINT D%; "PR#0"
3240 PRINT D%; "CLOSE"; FILE%
3250 VTAB 21: INVERSE : PRINT "-->"; NORMAL : PRINT
      "=>NEXT"; INVERSE : PRINT "<--"; NORMAL : PRINT
      "=<LAST"; TAB( 31); "RECORD #": VTAB 22: PRINT
      "PRESS"; INVERSE : PRINT "ESC"; NORMAL : PRINT
      : PRINT "FOR RECORD # "; TAB( 30); RX; " OF ";
      REC
3260 GOSUB 250: PRINT G%: CALL - 868: GET Y%: PRINT
      Y%: IF Y% = CHR% (3) THEN END
3270 IF Y% = CHR% (27) THEN VTAB 23: CALL - 8
      68: PRINT "ENTER RECORD NUMBER "; G%: INPUT
      RX: GOTO 3170
3280 IF Y% = CHR% (8) THEN RX = RX - 2: GOTO 33
      20
3290 IF Y% = CHR% (13) THEN 3140
3300 IF Y% = CHR% (21) OR Y% = CHR% (32) THEN
      3320
3310 GOTO 3250: REM ****
3320 RX = RX + 1: GOTO 3170
4000 REM CHANGE/DELETE DATA
4010 A1% = "CHANGE/DELETE DATA"
4020 TEXT : HOME : GOSUB 260: HOME : GOSUB 360:R
      X = 0: GOSUB 280
4030 INPUT REC
4040 PRINT D%; "CLOSE "; FILE%
4050 HOME : VTAB 23: HTAB 10: GOSUB 250: VTAB 5:
      PRINT : PRINT "ENTER RECORD # TO CHANGE/DEL
      ETE"; G%: INPUT RD%
4060 IF LEN (RD%) = 0 THEN CLEAR : GOTO 500
4070 RD = INT ( VAL (RD%)): IF RD < = 0 THEN D4
      050
4080 IF RD < = REC THEN 4100
4090 PRINT : PRINT "ONLY "; REC; " RECORDS ON FILE
      !"; PRINT : PRINT "PRESS RETURN TO CONTINUE.
      "; G%G%: INPUT Y%: GOTO 4050
4100 RX = RD: IF RX < = 0 THEN RX = 0
4110 GOSUB 280: FOR K = 1 TO NF: INPUT P%(K): NEXT
      K
4120 PRINT D%; "CLOSE"; FILE%
4130 GOSUB 310
4140 VTAB 22: CALL - 868: PRINT "CHANGE/DELETE
      RECORD # "; RD; " (Y/N) "; G%: GET Y%: PRINT Y
      %: IF Y% = "N" THEN 4050

```