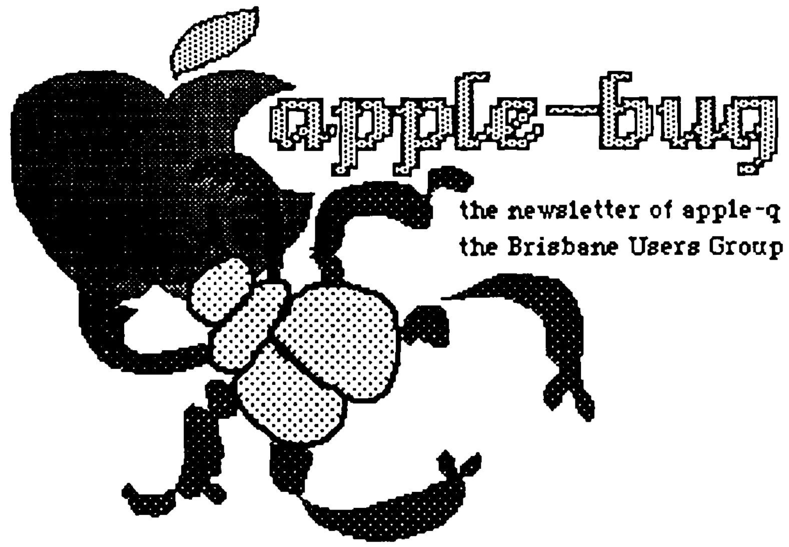


the newsletter of apple-q
the Brisbane Users Group

Apr 86

April 1986

In this issue:

Special Rush Issue - wer'e late again!

Mostly Tips & Techniques

Pending Book Review

CREDITS

Peter Pegg	Editor at large
Peter Newland	Editing in a Rush
Graham Hannam	Assistant Rush
MacWrite	word processing
LaserWriter	Printing and so on
Hooper Education Centre	printing and distribution

APPLE][, //e & //c the reason for it all!
and Apple ///, Lisa, Macintosh and Macintosh Plus too!

Published by Apple-Q, the Brisbane Users Group
P.O. Box 721, South Brisbane, Qld 4101 Australia
Registered by Australia Post - Publication No. QBG 3485

Editorial

Peter J. Pegg

First, special thanks to Peter Newland and Graham Hannam who are making sure that a newsletter is produced. I am finding that other matters which must, of necessity, be given the highest priority are leaving me little time for the normally enjoyable task of preparing apple-bug.

Recent news items show Apple Computer Inc. to be not only alive and well, but also flourishing with record earnings in the last quarter. The Macintosh has recently been enhanced as the Macintosh Plus, and a considerable amount of speculation is appearing in computer journals over the expanded future of the Apple II series. Some reports suggest that the II series will receive a SCSI port, and that the Macintosh and the II will one day share the same peripherals; other articles strongly suggest that future Apple IIs will have a Macintosh style interface. Obviously it is not yet possible to confirm these speculations, but they do seem to presage some very exciting developments. One message that does seem to be coming from Apple is that the II series is not dead yet, but has a glowing future. We all await this future with considerable interest if not outright enthusiasm.

What's when ...

Sunday 20 April 1986
Open day at the Hooper Centre

Monday 21 April 1986
Committee meeting

Sunday 18 May 1986
Open day at the Hooper Centre

Monday 19 May 1986
Committee meeting

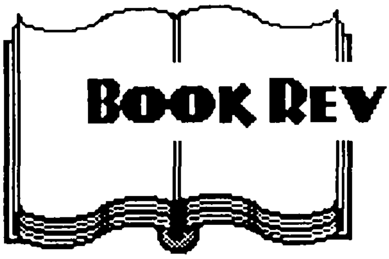
Notes

SECRETARY'S NOTES Graham Hannam

From time to time members have come up to the Trading Table requesting a power supply cord to connect their Apple to the Club's Kambrook board, because, in the rush to get to the meeting that particular item got left on the floor. Now we can come to your rescue as we have purchased a number of these items. The cost us $5 each and any member requiring one will be asked to pay a rental of $1 for the use of the cord and a refundable deposit of $4.

This is not to be confused with the 4-way power boards which are available at as a free service. We have purchased more power boards with overload protection switches built in.

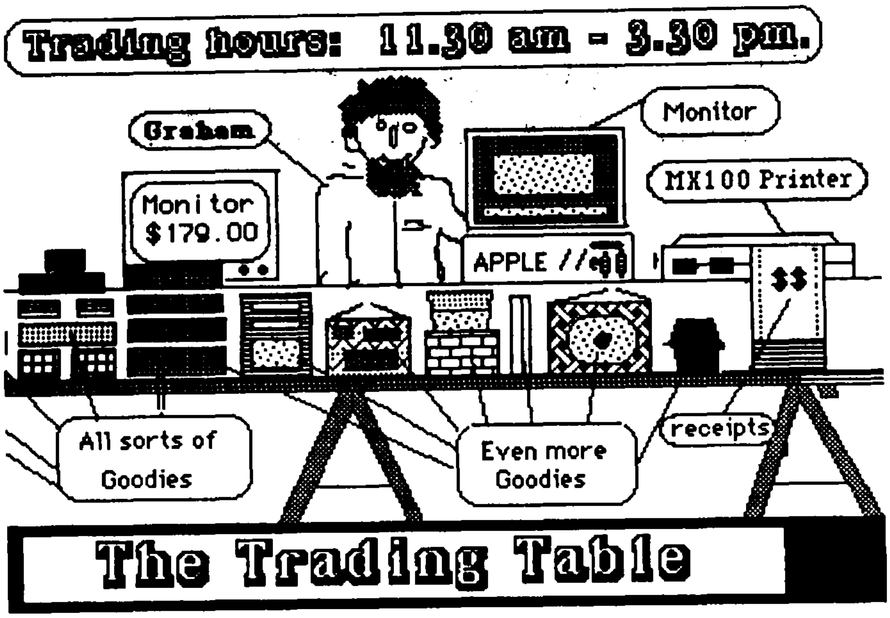
Due to an oversight in last months Newsletter we forgot to convey our appreciation to The Computer Shop, 137 Melbourne St., South Brisbane for the loan of an Apple //e with associated equipment to help run the trading table.



BOOK REVIEW DAVID BOURNE

Some new Apple related books have been received but as mentioned before David Bourne is on the other side of the world until sometime in June.

We hope to have the books reviewed by David as soon as he returns.



Due to unforeseen circumstances, the price of disks will rise considerably once our existing stocks are depleted. The Federal Government, in its wisdom, has imposed another tax on disks. The claim is that it is a mere 19% on the importers' costs, WHICH means more like 66% once it reaches the retail outlet. We have a reasonable stock at present, and our present selling price is less than our new cost price.

The Trading Table is the club's only means of raising funds, and we hope you will continue to suport the club in the future, as you have in the past.

Due to the continuing price changes, it has been an impossible task, trying to keep a price list up to date, and to publish it in Apple-Bug. All prices are confirmed only two days before the meetings, when the new stocks are purchased, so please bare with us. As you will see, many of the prices are unknown for the listed items, and therefore we cannot give a firm price.

If there is anything else you would like to see on the Trading Table, then please let us know and we will try and get it for you. We have simply run out of ideas as to what people would like to see on sale. We need your ideas.

Description		Price
Datalife SS/DD disks	(single)	$ 3.50
Datalife SS/DD disks	(lib.case)	$ 34.00
Control Data SS/DD disks	(lib.case)	$ 25.50
3.5" SS/DD disks	(softpak)	P.O.A
3.5" DS/DD disks	(softpak)	$ 75.00
Basic Programming	(book)	P.O.A
Workshops volumne 1	(book)	P.O.A
Workshops volumne 2	(book)	P.O.A
Computer Connection	(book)	P.O.A
Word Processing with ZARDAX	(book)	$ 16.00
Media drawers 3.5"	(each)	$ 28.00
DX-100A Media drawers 5.25"	(each)	$ 28.00
M.F. Media drawers 5.25"	(each)	$ 95.00
Mini disk tray .FD30	(each)	P.O.A
Epson MX/RX/FX ribbon refills	(each)	P.O.A
Epson MX80 ribbon cartridge	(each)	P.O.A
Epson LX80 ribbon cartridge	(each)	P.O.A
Epson LQ1500 ribbon cartridge	(each)	$ 16.00
Keyboard cover //e	(each)	$ 9.00
Keyboard cover //c	(each)	$ 9.00
Apple Keyring	(each)	$ 7.00

FROM THE TECH NOTES

SECRET PASSWORD

```
10    REM - SECRET PASSWORD
20    REM - THE USE OF MID$ FUNCTION,
30    REM - THE 'GET' STATEMENT,
40    REM - THE 'FOR...NEXT' LOOP, AND
50    REM - THE 'IF...THEN' STATEMENT.
60    REM
70    REM - RESET ALL VARIABLES.
80 CLEAR

85    REM - CLEAR THE SCREEN AND
86    REM - HOME THE CURSOR.
90 HOME

95    REM - THE PASSWORD IS IN PW$
100 PW$ = "APPLE"

110    REM
120    REM - BODY OF PROGRAM.
125    REM
130 PRINT "ENTER PASSWORD ";

135    REM - DETERMINE THE LENGTH OF PW$
140 FOR I = 1 TO LEN(PW$)
150 GET AN$: PRINT "*";

160    REM
170    REM - CHECK TO SEE IF KEY ENTERED
180    REM - IS PART OF THE PASSWORD (PW$)
190    REM - IN LINE 100.
200    REM
210 IF MID$ (PW$,I,1) = AN$ THEN X$ = X$ + AN$: NEXT I

220    REM
230    REM - CHECK TO SEE IF THE PASSWORD
240    REM - IS WRONG. IF IT IS THEN EXIT
250    REM - THE LOOP AND SET I TO LEN(PW$)
260    REM - AND THEN GO TO THE ERROR ROUTINE
270    REM - AT LINE 610.
280    REM
290 IF X$ < > PW$ THEN I = LEN(PW$): GOTO 610

300    REM - PASSWORD IS CORRECT.
310 HOME
320 PRINT "YOU GOT THE CORRECT PASSWORD !!!!!"
330 END

600    REM - PASSWORD IS WRONG.
610 HOME

612    REM - PRINT ON THE 10th LINE
615 VTAB 10
620 PRINT : PRINT "YOU BLEW IT !!!!!"

  625    REM - PRINT LINE 635 FIVE TIMES.
  630 FOR J = 1 TO 5
  635 PRINT CHR$(7) : REM BEEP
  640 NEXT J

  650    REM
  660    REM - GO BACK TO THE START OF
  670    REM - THE PROGRAM AND START
  680    REM - AGAIN.
  690 GOTO 10
```

COMMENTS:

This program will continue to run until you enter the correct password. The password is defined in line 100. If you wish to limit the number of attempts at the password, then insert the following FOR...NEXT loop at lines 82 and 690 :-

```
82 FOR K = 1 TO 5
690 NEXT K
```

< > means: DOES NOT EQUAL.

HOME means: CLEAR SCREEN and place the CURSOR in the top left hand corner of the screen.

CLEAR means: CLEAR ALL VARIABLES so that they equal nothing.

END means: END of program.

GET means: Single key input. The RETURN key is not needed.

LET means: Assign a value to a variable. LET A = 35. You do not need the LET command, so do not use it, as it takes up extra room in memory. LET A = 35 is the same as A = 35 in most computer languages. LET A$ = "COMPUTER" is the same as. A$ = "COMPUTER".

REM means: REMark. This is used to aid the programmer and not the program. The program will ignore the REM statement and everything that follows it in the same line. Get into the habit of using REM statements when writing a program.

LEN means: LENgth. LEN(PW$) will show the LENgth of the string PW$. If PW$ = "APPLE" then the LENgth of PW$ is 5 (the 5 characters within the quotes).

FORMULA PASSWORD

```
10     REM - FORMULA PASSWORD
20     REM
30     REM - RN = RANDOM KEY FOR
40     REM - FORMULA: RN * 4 - 2
50     REM -     "   : RN * CN - 2
60     REM
70     REM - RN$ = FORMULA IN LINE 160
80     REM
90 HOME
100 CLEAR
110     REM - GENERATE A RANDOM
120     REM - WHOLE NUMBER 1-9
130 RN = INT (RND (1) * 9) + 1

140     REM - SET FORMULA VARIABLES
150 CN = 4:CD = RN - 1

160     REM - SET PRINT POSITION
170 VTAB 6: HTAB 16

180     REM - CALCULATE THE ANSWER
185     REM - AND CONVERT RN TO A
190     REM - STRING VARIABLE
200 RN$ = STR$ (RN * CN - 2)

210     REM - GENERATE AND DISPLAY
220     REM - THE FORMULA KEY ALONG
225     REM - WITH SOME FALSE ONES
230 FOR C = 1 TO CN:CD = CD + 1: PRINT CD;"-";: NEXT C:
    PRINT CHR$ (8);" "
```

```
240     REM - ERAISE LINE 10 AND
250     REM - SET HTAB
260 VTAB 10: CALL - 868: HTAB 8
270 PRINT "ENTER PASSWORD ";

280     REM - SETUP LOOP TO ENTER
290     REM - PASSWORD
300 FOR I = 1 TO LEN (RN$)

310     REM - ENTER PASSWORD AND PRINT
320     REM - A SPACE AND A BACKSPACE
330 GET AN$: PRINT " "; CHR$ (8);

340     REM - IF THE ENTRY IS CORRECT
350     REM - GO BACK AND GET THE NEXT
360     REM - CHARACTER
370 IF MID$ (RN$,I,1) = AN$ THEN X$ = X$ + AN$: NEXT I

380     REM - IF THE PASSWORD IS
390     REM - CORRECT, THEN CONTINUE
400     REM - PROGRAM
410 IF X$ = RN$ THEN PRINT: GOTO 500

420     REM - GO BACK TO LINE 90
430 GOTO 90
440 END

500 VTAB 22: HTAB 8
510 PRINT "CORRECT CALCULATION"
```

COMMENTS:

I know that password programs are a dime a dozen, but here is one with a difference. The formula in this program can be changed in several ways. You can change the calculation, and you can change the random key. You need remember only the formula, and when the program displays the numbers, you calculate the first number (RN * CN - 2) and enter the answer. The additional numbers are only there to confuse anyone who would try to guess your password.

CHANGING PASSWORD

```
10     REM - CHANGING PASSWORD
20     REM
30     REM - THE NUMBER OF PASSWORDS
40     REM - IS SET IN THE FIRST DATA
50     REM - STATEMENT IN LINE 140 AND
60     REM - THE PASSWORDS THEMSELVES
70     REM - ARE IN LINE 160
80     REM
90     REM
100 HOME

110     REM - SET UP ARRAY FOR
115     REM - NUMBER OF PASSWORDS
120 DIM PW$(11)

130     REM - NUMBER OF PASSWORDS
140 DATA 11

150     REM - PASSWORDS
160 DATA ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,NINE,TEN,FINISH

170     REM - READ FIRST DATA ELEMENT
180 READ A

190     REM - READ PASSWORDS AND PUT
200     REM - THEM INTO AN ARRAY
210 FOR I = 1 TO A: READ PW$(I): NEXT I
```

```
220    REM - ERAISE 10TH LINE
230 VTAB 10: CALL - 868
240 PRINT "ENTER PASSWORD ";

250    REM - NUMBER OF ATTEMPTS AT PASSWORD
260 FOR P = 1 TO A

270    REM - FIND LENGTH OF PASSWORD
280 FOR I = 1 TO LEN (PW$(P))

290    REM - ENTER A CHARACTER AND PRINT *
300 GET AN$: PRINT "*";

310    REM - IF CHARACTER IS CORRECT
320    REM - GO BACK AND GET THE
325    REM - NEXT CHARACTER
330 IF MID$ (PW$(P),I,1) = AN$ THEN X$ = X$ + AN$: NEXT I

340    REM - IF CHARACTER IS WRONG, EXIT LOOP
350    REM - AND GO TO ERROR ROUTINE
360 IF X$ < > PW$(P) THEN I = LEN (PW$(P)): GOTO 5020

370    REM - PASSWORD IS CORRECT
374    REM - RESET P AND EXIT LOOP
376 P = A: NEXT P
380 PRINT : VTAB 20
390 PRINT "YOU GOT THE CORRECT PASSWORD"

395    REM - END PROGRAM
400 END

5000    REM - ERROR ROUTINE
5010    REM - SET X$ TO NULL STRING
5020 X$ = "": PRINT
5030 VTAB 20

5040    REM - GET NEXT DATA STATEMENT
5050 READ BL$

5060    REM - PRINT TEXT, DATA, AND BELL
5070 PRINT "YOU HAVE BLOWN YOUR ";BL$;" ATTEMPT";CHR$ (7)

5080    REM - PAUSE ROUTINE
5090 FOR F = 1 TO 1000: NEXT F

5100    REM - CLEAR FROM 10TH LINE TO
5105    REM - BOTTOM RIGHT HAND CORNER OF SCREEN
5110 VTAB 10: CALL - 958: HTAB 16

5120    REM - GO BACK AND ATTEMPT NEW PASSWORD
5130 NEXT P
5145    REM - DATA FOR BL$
5150 DATA 1ST,2ND,3RD,4TH,5TH,6TH,7TH,8TH,9TH,10TH,LAST
```

COMMENTS:

This program has many passwords. If you make a mistake entering the password, then the password changes, and you can try again. Now you need to remember a whole list of passwords, and their correct order.

PAUSE ROUTINES

Any one of these routines can be inserted where you want a program to pause.

```
100 PRINT "PRESS ANY KEY TO CONTINUE ";: GET AN$: PRINT
```

```
100 INPUT "PRESS RETURN TO CONTINUE ";AN$
```

I don't know about you guys but I am always looking for some new trick for my programs. I have been reading the manuals and old magazines and generaly picking the brains of other computerists for 'TIPS & TECHNIQUES' and here are a few for your perusal:

PEEKs POKEs & CALLs

CALL -155	enter monitor: no bell
CALL -151	enter monitor: with bell
POKE 44033,16	prevent <CATALOG>
POKE 2049,1	LIST protection: 1st line continually listed
PEEK(218) + PEEK(219) * 256	line number where ERROR occurs
X = PEEK(-16287)	read push button on PDL # 0: IF X > 127 THEN button pressed
X = PEEK(-16286)	read push button on PDL # 1: IF X > 127 THEN button pressed
X = PEEK(222)	looks at ERROR code page 136. REFERENCE MANUAL
X = PEEK(37)	reads the current vertical
position	
	of cursor: POKE 37,X
100 X = PEEK(-16384)	read keyboard strobe: IF X > 127 THEN key pressed
110 POKE -16368,0	reset keyboard strobe
CALL -958	clear from cursor to bottom of screen
POKE 32,L	set left margin : L = 1 to 39
POKE 33,W	set right margin : W = 1 to 39
POKE 34,T	set top margin : T = 1 to 23
POKE 35,B	set bottom margin: B = 1 to 24
CALL 65200	NEW
CALL -1998	fill screen with INVERSE @
CALL 1002	reconnect DOS

RECOVER (from disaster)

Have you ever deleted a program by typing NEW or INT accidentally? It happens to the best of us at the most inconvenient times. Here is a good little machine language program that will restore it for you. To use it, BRUN RECOVER from your disk directly after your accident. When you type LIST your old program will be back again.

NOTES:

1> Only APPLESOFT programs can be recovered.
2> If you typed INT you must type FP to return to APPLESOFT before BRUNing RECOVER.
3> To execute the program just BRUN RECOVER or BLOAD RECOVER and CALL 768.

```
0300- A2 00 86 06 A2 08 86 07
0308- A0 05 B1 06 C8 D0 02 E6
0310- 07 C9 00 D0 F5 8D 00 08
0318- 8C 01 08 A6 07 8E 02 08
0320- A2 02 D0 02 A2 03 B1 06
0328- C8 D0 02 E6 07 C9 00 D0
0330- F3 CA D0 F2 84 69 84 AF
0338- A6 07 86 6A 86 B0 60
```

```
BSAVE RECOVER,A$300,L$3E
```