Brafune

mini'app'les

apple computer user group newsletter

MAY 1979 VOL II No 4

Daniel B.Buchler, President & Activity Coordinator 890-5051
Chuck Thiesfeld, Treasurer & Newsletter Editor (830-5020 (884-8208
Chuck Boody, Secretary 873-2227
Keith Madonna, Librarian 474-3876
Rob Wentworth, Program Editor 825-9086
James Henke, Technical Advisor 869-6371
Deän Anderson, Bibliographer 466-5562

Contributions
Complaints
Correspondance
D.Buchler
13516 Grand Ave S
Burnsville, Mn
55337

Membership
C.Thiesfeld
8610 Russel Ave.S
Bloomington
Mn 55431

NEXT MEETING

WEDNESDAY MAY 16, 1979 7:30 P.M.
MINNESOTA FEDERAL SAVINGS & LOAN, 31 9TH AVENUE SO. HOPKINS MN.

HOW WOULD YOU LIKE TO RUN YOUR PROGRAMS 10 TIMES FASTER? "IT CAN'T BE DONE"YOU SAY. IT CAN BE DONE AND SOME PROGRAMS WILL BE FASTER THAN THAT. LEARN HOW YOU CAN DO IT AT THE NEXT MEETING. DAVE LARSON AND CHUCK THIESFELD WILL DEMONSTRATE XPLO. XPLO IS A HIGH LEVEL LANGUAGE MUCH LIKE PASCAL. AHH YOU'VE HEARD ABOUT PASCAL. WELL RUMORS HAVE IT PASCAL WILL COST $300-$500, XPLO IS ONLY $35 AND RUNS ON YOUR 32K SYSTEM DISK OR TAPE. DAVE AND CHUCK WILL EXPLAIN STRUCTURED PROGRAMS, XPL0 SYNTAX AND STRUCTURE. INTRINSICS CAN BE WRITTEN FOR ALL YOUR SPECIAL FUNCTIONS, EVEN FLOATING POINT. HERE'S A CHANCE FOR YOU BEGINNERS TO GET IN ON THE FIRST LESSON. DONT MISS YOUR CHANCE TO GET STARTED WRITING BETTER PROGRAMS NOW.

------ LETTER FROM THE EDITOR:
I WANT TO TAKE THIS OPORTUNITY TO SAY THIS CLUB HAS MADE A VERY GOOD START AT HELPING IT'S MEMBERS. THE MAIN PURPOSE OF THE CLUB IS TO BRING PEOPLE WITH A COMMON INTEREST (APPLES OF COURSE) TOGETHER. AND WHEN THAT HAPPENS THEY SHARE EXPERIENCES, PROBLEMS AND ACCOMPLISHMENTS. TO THAT END SO DOES YOUR NEWSLETTER STRIVE. SO SHARE YOUR PROGRESS WITH EVERYONE IN THE NEWSLETTER. WRITE ABOUT YOUR PROGRAMS YOUR NEEDS, YOUR DREAMS. TO MAKE IT EASIER FOR YOU AND ME,I PRODUCE THE NEWSLETTER ON THE APPLE. I WILL ACCEPT ARTICLES IN ANY FORM BUT THE BEST WAY IS BY CASSETTE (WHICH WILL BE RETURNED BY THE WAY). A SIMPLE BINARY DUMP OF ASCII CODES IS TH BEST. A QUICK PROGRAM LIKE THE FOLLOWING CAN BE USED TO STORE IT.

```
10  CALL-936:CNT=0:BASE=4096:  DIM  L$(80)
20 INPUT L$
30 FOR A=1 TO LEN(L$)-1
40  POKE  A+CNT+BASE,ASC(A,A):NEXT  A
50 CNT=CNT+LEN(L$)
60 IF CNT>2048 GOTO100
70 GOTO 20
100  PRINT  "RESET  THEN  1000.1800W"
120  PRINT  "MAIL  TO  C.W.THIESFELD"
130 PRINT "8610 RUSSELL AVE."
```

A PROGRAM WILL BE PUT ON THE USER BANK WHICH CAN EDIT WITH UPPER AND LOWER CASE IF YOU WANT TO GET REALLY FANCY.
BUT THE POINT IS REALLY ----- SHARE YOUR SUCCESS AND SORROWS.------ YOU'LL FEEL BETTER.

THE MINUTES FOR THE BUSINESS MEETING OF APRIL 18 1979

THE MEETING WAS CALLED TO ORDER AT 7:45 BY THE CHAIRMAN DAN BUCHLER.

THE FLOOR WAS OPENED FOR NOMINATION FOR OFFICERS. THERE WERE NO NOMINATIONS FROM THE FLOOR. A MOVE WAS MADE TO ELECT THE OFFICERS BY ACCLAMATION BY KEVIN AND SECONDED BY KEN SLINGSBY. THE MOTION WAS CARRIED BY UNANIMOUS APPROVAL. THE OFFICERS ARE PRESIDENT - DANIEL BUCHLER, SECRETARY - CHARLES BOODY, AND TREASURER - CHARLES THIESFELD.

THE PRESIDENT CALLED FOR NOMINATION FOR BOARD MEMBERS TO BE ACCEPTED FOR THE POSITIONS OF LIBRARIAN, PROGRAM EDITOR, TECHNICAL ADVISOR, NEWSLETTER EDITOR, BIBLIOGRAPHER, AND ACTIVITY COORDINATOR. A MOVE WAS MADE TO ACCEPT KEITH MADONNA, ROB WENTWORTH, JIM HENKE, CHARLES THIESFELD, DEAN ANDERSON AND DAN BUCHLER RESPECTIVELY FOR THE ABOVE POSITIONS BY ACCLAMATION BY PHIL SHULER. THE MOTION WAS SECONDED BY STAN BROOKS AND CARRIED BY UNANIMOUS DECISION.

A SUGGESTION WAS MADE FOR NAME TAGS FOR FUTURE MEETINGS.

IT WAS ANNOUNCED THAT MECC CAN BE ACCESSED AND THEY NOW HAVE A DOWN LOADER FOR THE APPLE. THE COST IS $50.00 FOR 50 HOURS IN 3 MONTHS. THE PERSON TO CONTACT IS: DON RAWITSCH AT 2520 BROADWAY 376-1101.

A SUGGESTION WAS MADE FOR A PROJECT TO COLLECT SUBROUTINES FOR TRADE. IF YOU HAVE A SUITABLE ROUTINE CONTACT THE LIBRARIAN.

AN ANNOUNCEMENT WAS MADE THAT THE CLUB HAS PERMISSION TO REPRODUCE THE WOZPAK AND $8.00 PER COPY WILL BE COLLECTED.

DUES WERE APPROVED AT $10.00 PER YEAR.

THE PRESIDENT RECOGNIZED COMPUTERLAND AND THANKED THEM FOR THEIR TREMENDOUS HELP AND SUPPORT.

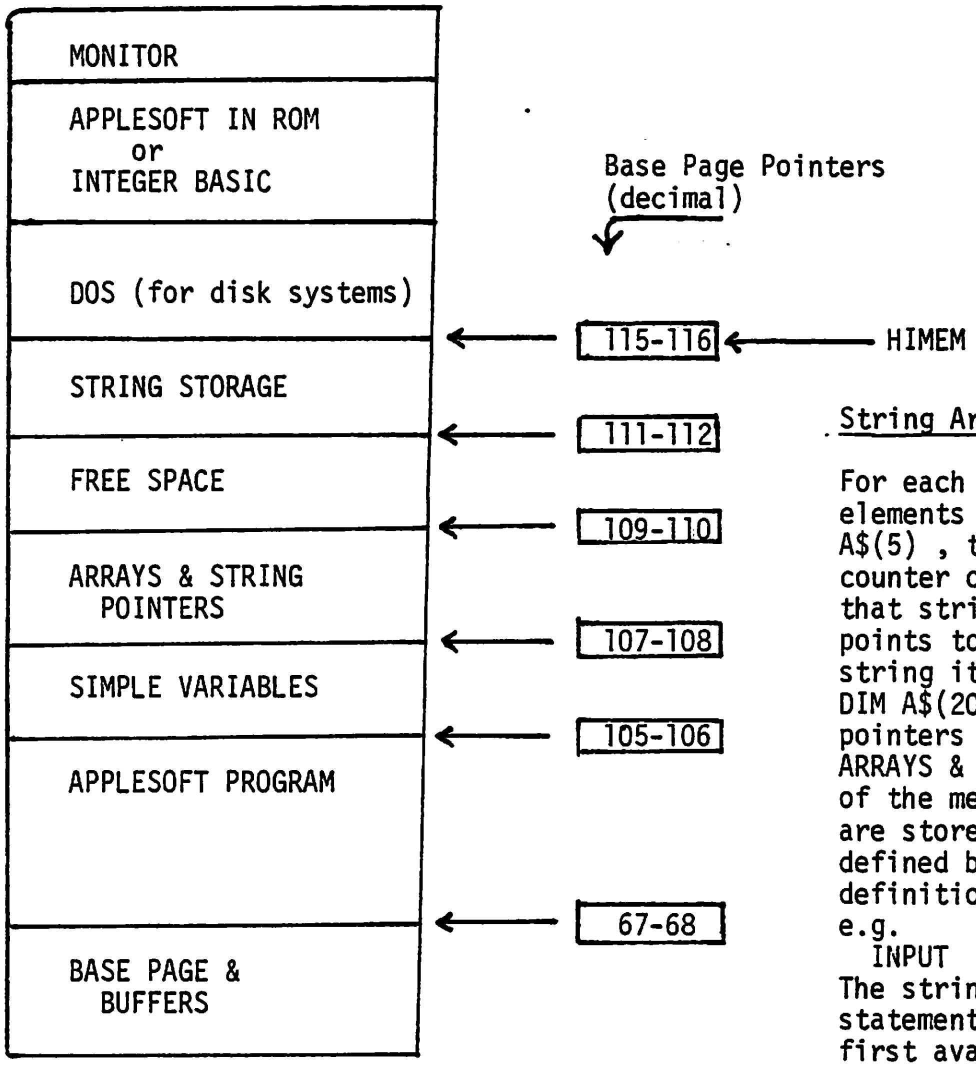
IT WAS SUGGESTED THAT THE BEGINNERS HAVE A MEETING. THE TECHNICAL ADVISOR JIM HENKE CAN BE CONTACTED AT 869-6371

KEITH MADONNA AND ROB WENTWORTH GAVE A LIST OF THE CURRENT LIBRARY PROGRAMS.
THE MEETING WAS ADJOURNED TO BEGIN THE PROGRAM TRADING SESSION AT 8:20

WOZPAK THE WOZPAK MASTER WAS RECEIVED FROM CALL A.P.P.L.E AND IS IN REPRODUCTION AT THE TIME OF THIS WRITING. WE HOPE TO HAVE IT READY FOR DISTRIBUTION BY THE NEXT MEETING. WE MAY HAVE ONE OR TWO EXTRA COPIES BUT ANYONE WHO WANTS ONE WHO DID'NT ORDER ONE, WILL HAVE TO WAIT FOR NEXT PRINTING OR ORDER ONE FROM A.P.P.L.E. -

DATA STORAGE IN APPLESOFT

Manipulation of data blocks in Applesoft (discussed elsewhere in this issue) require an understanding of the storage organisation of Applesoft.



String Arrays:

For each string (including elements of a string array,e.g. A$(5) , there is a pointer and counter of number of characters in that string.The pointer, 2 bytes, points to the start of the string itself. A DIM statement, e.g. DIM A$(20) reserves space for the pointers and counters in the ARRAYS & STRING POINTERS area of the memory. No string characters are stored until a string is defined by execution of such a definition within the program.
e.g.
```
  INPUT  A$(5)
```
The string input by the above statement will be stored in the first available space in STRING STORAGE (see figure).

The STRING POINTER in the ARRAYS & STRING POINTERS area will be updated to point to the string, and the counter will contain the length of the string (1 byte). Also the Base Page Pointer in 111-112 will be updated to point to the first character of the string which is at the bottom of the STRING STORAGE space.

Simple String Variables

Simple Strings(non dimensioned strings) are treated in a similar way to string arrays. Each simple string consists of a 7 byte field stored in the simple variable space. Within that 7 byes are the string name and a pointer/counter identical in format to that of the string array elements.

Arrays

Space is reserved for the Arrays in the ARRAY and STRING POINTERS space on execution of a DIM statement
```
          e.g.  DIM  A(100)
```
reserves space for 100 floating point (real) numbers of 5 bytes each. The array is added to the top of the area ARRAY & STRING POINTERS, and the base page pointer 109-110 is updated to point to the next available byte above that area.

DATA STORAGE IN APPLESOFT (Cont)

Simple Variables

Space is reserved for simple variables in the SIMPLE VARIABLES area.(See figure above) As simple variables are encountered in the program, they are added to the top of the SIMPLE VARIABLES space. Also the base page pointers 107-108 and 109-110 are updated to point to the start of THE SIMPLE VARIABLES space and FREE SPACE accordingly.

Note whenever a new statement is added to a program, both the simple variables, and the ARRAY & STRING POINTER space must move up in memory to make room for new statements. In actual fact, changing the size of the program in APPLESOFT will be cause for initialization of all the base page pointers

Note:

SPACE UTILIZATION IN APPLESOFT

Type	Example	Bytes required
INTEGER VARIABLES	I%	7
REAL VARIABLES	X	7
INTEGER ARRAYS (1 dimension)*	A%(10)	7 + 2n
REAL ARRAYS (1 dimension)*	B(20)	7 + 5n
SIMPLE STRINGS	C$	7 + c
STRING ARRAYS	D$(200)	7 + 3n + c

Key

*Add 2 bytes for each additional dimension

n = total number of elements in array.

c = total number of characters relating to each simple string or the whole string array.

NOTE on STRINGS. Use of a literal
eg A$ = "THIS IS A LITERAL"
will not store anything in the string space. The pointers in the ARRAY & STRING POINTERS area will point to the literal char ters in the program ~~itself~~.

UPPER/LOWER CASE EDIT

In the April MINI'APP'LES newsletter, we reported on the Wentworth subroutine package which creates characters of your choice in the HIRES mode. That package slightly modified, in conjunction with an APPLESOFT INPUT routine, provides the capability to upgrade almost any BASIC text edit package to full ASCII . That is, the EDIT package will be able to output ASCII codes for upper and lower case letters and one can view the characters as upper and lower case.

The Game button #'0' is used as the shift key. The output of the EDIT package may then be printed on printers which provide upper and lower case characters.A full description of the modified Wentworth software and the input routine will appear in next month's newsletter.

The abovesoftware has been incorporated in a mini-EDIT program the function of which is to allow pursons who wish to provide documentation for the newsletter a means of creating it on their Apples. Besides allowing upper/lower case manipulation, the output is a text file on cassette tape. (could easily be changed to a disk file). The output is then entered into the more general word processing editor which we plan to use for the newsletter.

LOADING MACHINE CODE SUBROUTINES - (Idea courtesy Val Golding, et al)

The guys from A.P.P.L.E. have recently adopted an interesting and very powerful technique for executing Monitor functions with complex cammand lines, right in the middle of an APPLESOFT program. In particular this allows an easy way to input or load machine code that goes with your Basic program.

Define your machine code or Monitor command sequence just as if you were using the Monitor, only instead of tying the line follwing a * prompt character, inclose the line in quotes as a literal string.

Example 1 below loads the hex data C9 48 etc into location 300, 301 etc

```
310 A$ = " 300:C9 48 D0 12 20 B1
     00 20 F8 E6 8A 20 DA FD 20 B
     7 00 C9 2C F0 EF 60 4C CC 16"

350 A$ = A$ + " N D823G": REM

400  FOR I = 1 TO  LEN (A$): POKE
     511 + I, ASC ( MID$ (A$,I,1)
     ) + 128: NEXT : POKE 72,0: CALL
      - 144
410  GOTO 36
```

Example 2 executes a Monitor Move command

```
410 A$ = "800<400.7FFM"
```

Note : In statement 350, the N (a monitor 'Normal Text mode' command delimits(ends) the list of bytes for storing in memory; the D823G is an entry point in Applesoft which clears the stack and returns control to the Applesoft program at the next statement following the CALL -144. In this case, statement 410, if the D823G is not included in the literal string, control returns to the Monitor and you end up with an * prompt character ! Because the stack is cleared, the CALL -144 must not be inside a subroutine called by a GOSUB.

NEW PRODUCTS/NEW IDEAS

Micromodem II D.C.Hayes of Atlanta Ga. have just announced a combined Modem and Interface card for the Apple. The combination consists of a card that plugs into an Apple slot and a small box connected by cable to the card.. The combination provides:

- °Bell 103 Data set compatible transmission at 110 or 300 baud.
- °Direct connect to phone line using the RJ11 type jack commonly found on Western Electric manufactured phones and handsets.
- °Firmware in ROM on the card.
- °Auto-Answer and Auto-Dial

The price is around $380 which actually compares favourably with price of Apple Comm card and currently available quality modems. In fact the Auto-Answer/Auto-dial would normally cost you alot more.

Homemade If the $380 above scares you, and you want communications on your system, look at the article in April 1979 Byte magazine entitled "Cross Pollinating the Apple II" by R.Campbell. It describes the construction of an RS-232 compatible programmable interface based on the INTEL 8251 chip. The device can be built for about $25 in parts not including the P.C. board. A prototyping board with good documentaion is available from Apple or you can save money by buying a similar board from some other sources who offer boards that are plug compatible with the Apple slot.(see below)

Printers Heathkit are now offering a good quality dot matrix printer in kit form for $600 (Builtup $900). It has specifications similar to the Brightwriter IP-225 including tractor feed. It does not have graphic option but has some other features. It has a serial interface which will connect to Game Port via a 3 part lashup.

Radio Shack have a cheap Thermal print- $220. But its only 36 columns. Has serial interface too.

Concluded on bottom of next page, and page 9

APPLESOFT II ALL ENCOMPASING TAPE DATA SAVE

There is no direct way in APPLESOFT to save strings on casette tape. Contact #3 described a method which writes data on tape as follows.

RECORD 1 Length of string area
RECORD 2 Pointers for one string array (USES STORE statement)
RECORD 3 All of String Storage

This works fine if everything to be saved is in one string array. However, typically one will wish to save alot of string arrays, and simple string variables. The Applesoft manual tells one to convert strings to numerical data and use the STORE statement. This is not practical with large amounts of string storage. I just completed a program in which it was necessary to save the following:

4 string arrays
2 numeric arrays
12 simple varaibles

Using the technique described in CONTACT #3, and assuming the simple variables were stored in a 3rd numeric array, 9 records would be required to output the data as follows:

1 RECORD . Size of string area
4 RECORDS of string pointers
3 RECORDS of numeric arrays
1 RECORD of the string area

Because the tape records each include the standard long 'leader', the I/O time would be excessive.

The solution is to save all of the 'data' in one fell swoop. Refer to data organization chart in accompanying article on Applsoft storage organisation. For efficiency one does not want to save the unused part of memory (FREE SPACE). This could result in very long records for systems with large memory sizes(48k). Therefore three records are required:

RECORD 1 length of simple variable/array & string pointer space
RECORD 2 simple variables; array & string pointer space; address of start of string space; length of string space
RECORD 3 string space

The following program will output all of the data in the above format. Also, the program may change between output and inputting the data provided certain restrictions are observed. Those restrictions are:

1. HIMEM cannot change. This is because the string pointers are absolute adresses.
2. If the DIM statements are changed, no effect will be observed since the input data redefines all data space . DIM statements executed after input will of course be observed.
3 There must be at least 4 bytes of FREE SPACE to store the pointer and length inf in record 2.

NOTES: This program uses DEF statements as described elsewhere in newsletter. The complex code with lots of PEEKS and POKES was necessary to avoid referencing variables during the redefinition of pointers. A slightly more efficient technique would involve finding 6 unused bytes in the base page somewhere . Anyone want to try ? (Program listing next page)

NEW PRODUCTS/NEW IDEAS (Continued from previous page)

Apple II Delicacies California Computer Systems has a series of boards (delicacies) that plug into an Apple slot. Included are: a 2716 PROM board, Programmable Timer , IEEE 488 Interface, Asynchronous Serial Interface with ROM ($100), Synchronous Serial Interface with ROM, Parallel Interface with ROM, Arithmetic Processor with 32 bit floating point capability-uses AMD 9511 chip ($400), 3.5 digit BCD A-D converter, solder tail board ($21), Wire Wrap Board, and Etch board.

APPLESOFT ALL ENCOMPASING TAPE DATA SAVE (Continued)

SUBROUTINE TO OUTPUT TO TAPE

Note: PB=I+PA makes sure that these variables are allocated before pointers are read.

```
28000 PB = I + PA:PA =  FN Q(109)
     -  FN Q(105) + 4
28005  PRINT "PA= ";PA
28010  GOSUB 29100
28020  POKE 30, FN M(PA): POKE 31
    ,PA / 256: GOSUB 29010: REM
     WRITE LNGTH VARS & ARRAYS
28030 PA =  FN Q(109)
28040  POKE PA + 2, PEEK (111): POKE
     PA + 3, PEEK (112)
28050 PB =  FN Q(107) -  FN Q(105
    ): REM  LENGTH SIMPLE VAR SP
    AVE
28060  POKE PA, FN M(PB): POKE PA
     + 1,PB / 256
28070  GOSUB 29040
28080  PRINT  FN Q(105);" TO ";PA
     + 4
28090  GOSUB 29010: REM  WRITE VA
    RS & ARRAYS
28100  GOSUB 29020
28110  PRINT "STRINGS START AT ";
     FN Q(111)
28120  GOSUB 29010: REM  WRITE OU
    T STRINGS
28130  PRINT "OUTPUT";
28200  PRINT " DONE; FREE SPACE =
     "; FRE (Y$): PRINT "TERMINA
    TE ?": GOSUB 20100: END
29000  POKE 60,30: POKE 61,0: POKE
    62,31: POKE 63,0: RETURN
29010  PRINT "WRIT";: GOSUB 29013
    : CALL  - 307: RETURN
29013  PRINT "ING RECORD #";I;:I =
    I + 1
29015  PRINT " - LENGTH = "; FN Q
    (62) -  FN Q(60) + 1;" BYTES
    ": RETURN
29020  POKE 60, PEEK (111): POKE
    61, PEEK (112): POKE 62, PEEK
    (115): POKE 63, PEEK (116): RETURN

    : REM   SET POINTERS FOR STRING S
    PACE
29030  PRINT "READ";: GOSUB 29013
    : CALL  - 259: RETURN
```

NOTE: Subroutine 29300 must be CALLed prior to CALLing either 28000 or 29500. This initializes the DEF . See article SIMPLE FUNCTIONS in APPLESOFT elsewhere in this newsletter.

```
29040  POKE 60, PEEK (105): POKE
    61, PEEK (106): POKE 62, FN
    M(PA + 4): POKE 63,(PA + 4) /
    256: RETURN
29100  GOSUB 20000: PRINT :I = 1:
     PRINT "MOUNT TAPE; ARE YOU
    READY?": GOSUB 20100: GOSUB
    29000: RETURN
29300  DEF  FN M(P) = P -  INT (P
     / 256) * 256
29310  DEF  FN Q(P) =  PEEK (P) +
    256 *  PEEK (P + 1)
29320  RETURN
```

SUBROUTINE TO INPUT FROM TAPE

Note PA=PB +I makes sure these variables are allocated before reading pointers.

```
29500 PA = PB + I: GOSUB 29100: GOSUB
     29030: REM  READ LGTH REC
29520 PA =  FN Q(30) +  FN Q(105)
     - 4
29530  GOSUB 29040: REM  PTRS FOR
     VARS $ ARRAYS
29540  POKE 109, FN M(PA): POKE 1
    10,PA / 256
29545  GOSUB 29030
29565  POKE 111, PEEK ( PEEK (109
    ) + 256 *  PEEK (110))
29566  POKE 112, PEEK (1 +  PEEK
    (109) + 256 *  PEEK (110))
29567  POKE 107,( PEEK (105) +  PEEK
    (111) + 256 * ( PEEK (106) +
     PEEK (112))) -  INT (( PEEK
    (105) +  PEEK (111) + 256 *
    ( PEEK (106) +  PEEK (112)))
     / 256) * 256
29568  POKE 108,( PEEK (105) +  PEEK
    (111)) / 256 +  PEEK (106) +
     PEEK (112)
29570  GOSUB 29300:PA =  FN Q(109
    ):PB =  FN Q(PA + 2)
29574  PRINT "STRING PTR PB= ";PB

29580  POKE 111, FN M(PB): POKE 1
    12,PB / 256: GOSUB 29020: GOSUB
    29030: REM  STRING SPACE
29590  PRINT "INPUT";: GOTO 28200
```

— — — — — — — — —

```
20100 PRINT "TYPE Y OR N";: GET W$:
      IF W$ = "Y" THEN RETURN
20110 REM CONTINUE ON IN MAIN PROGRAM
20120 POP:    etc etc
```

SUMMARY OF ARTICLES IN Call A.P.P.L.E.

In past issues of this newsletter, we have reproduced whole articles from, or extracted from the Call A.P.P.L.E. newsletter. In fact there are some ideas extracted from A.P.P.L.E in this issue. In as much as that newsletter is one of, if not the source of latest techniques, ideas and information on the Apple, it was thought appropriate to include a summary of the lead articles in our own newsletter. We do have a newsletter exchange with Seattle. Thus if anyone sees something that might be of interested they can borrow the appropriate issue from our Librarian or whoever has one. All of you might consider getting your own subscription, which costs $7.50 per year. It is an excellent publication.

Call A.P.P.L.E. Feb issue:

The Talking Apple by Mark A.Cross --- Describes a speech synthesis method which utilizes some very simple hardware interfaced to the Game port. It is supposedly superior to Appletalker(Bob Bishop) and as inteligible as a good tape recorder. Addendum in Mar issue.

Chaining Applesoft (Cont from Jan edition) by Randy Wiggington

Disk-to-Disk Transfer by Ron Aldrich

INteger Basic Entry Points by Val Golding --Describes a program which will display entry points for various tokens., the entry points being addresses in the BASIC ROM.

Call A.P.P.L.E March issue:

Applesoft from Bottom to Top by Val Golding---5+ pages of detail discussions on memory mapping;a routine to create a Dispatch Table List; how to Append; and how Applesoft pointers are used.

The Mystery of Text Files by Darrell Aldrich---Discusses sequential & random access.

Disk Access Utility by Don Paymar-- A program to dump entire contents of a disk, track by track, sector by sector to display.

Applelock by Dick Sedgewick --- Technique to write protect or lock a disk program.

Keyboard Modification to get"[","\" or "_" characters by Dan Paymar---- Shows how to modify P.C. board under keyboard to provide keyboard generation of above characters with SHIFT K, SHIFT L and SHIFT D respectively.

Call A.P.P.1.E has instigated a Cassette of the month plan in which $3.50 buys a cassette from Call A.P.P.L.E. containing whatever new programs are available that month. We Mini'App'Les are thinking of subscribing on behalf of our bank. Any comments ?

PRINTING WITH TABS IN APPLESOFT

A session with a printer connected to an APPLE SERIAL INTERFACE card revealed some interesting problems related to the use of TAB with APPLESOFT. A particular Applesoft program includes code to select the order in which fields are printed according to the value of the variables used as argument of the TAB function.

```
100 PRINT   TAB(A) ;  "FIELD 1";    GO TO  next line
200 PRINT   TAB(B) ;  "FIELD 2";    GO TO  next line
```

The logic of the program is such that

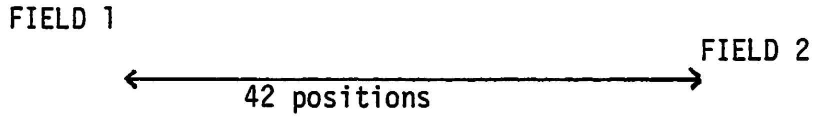
```
500 IF  A < B  THEN   100
600 GO TO  200
```

In a particular case, A = 1 , B = 42. On the standard TV display, we see

```
FIELD 1
 FIELD 2
```

PRINTING WITH TABS IN APPLESOFT (continued from previous page)

Applesoft Tabs assume a string length of 255 characters (maximum), but a line length of 40, so the TAB (42) results in a second line. .
The Serial Interface card has a DIP switch on which one can select certain standard line lengths. The setting for 80 cols was used and the resulting output for this example was



The manual for the card does state that 'there are certain restrictions on the use of tabs'. What apparently happens above is that the C/R from the end of the first 40 character line is converted to a Line Feed by the firmware !

The serial interface card firmware also makes use of memory location 1785 for the line length (for slot #1)

POKE 1785,80 sets the line length. In this example, the message

OUT OF MEMORY ERROR at stmt no

was encountered on use of the tab feature so long as the tab variable value was greater than 40 - I did'nt try every combination.

The moral of the story is'be careful how you ses TAB in Applesoft with printers on the serial interface board'. Perhaps use of the SPC function might solve the above problems.

SIMPLE FUNCTIONS IN APPLESOFT.

Few Applesoft programs seem to use the DEF capability. One neat application of DEF is to reduce the code needed to handle common usages of PEEKS and POKES. Two examples of this type of application are

1. Simulate the INTEGER BASIC MOD function

```
10 DEF  FNM(X) = X -     INT( X / 256 ) * 256

50 POKE  105, FNM(A)   : REM PUTS LOW  ORDER BYTE OF INTEGER PORTION OF A IN 105
60 POKE  106,  A/256   : REM HIGH ORDER BYTE OF INTEGER PART OF A IN 106
```

The above are equivalent to the INTEGER BASIC statements:

```
50 POKE 105, A MOD 256
60 POKE 106, A/256
```

2. Access 2 byte (16 bit) address or data in 2 memory locations

```
20 DEF FNQ(Y)= PEEK ( Y ) + 256 * PEEK( Y + 1 )


50 PRINT  FNQ(105)  : REM PRINTS 16 BIT  address or data stored in 105 & 106
```

NEW PRODUCTS/IDEAS (Continued from page 6)

Clock/Proto-board West Side Electronics, Chatsworth, Ca. offers the APPLETIME clock board. It can utilize internal,AC or battery power and sells for $60. They also list a prototyping board for $17.

HOW MANY APPLES

ACCOORDING TO DATAMATION MAGAZINE, PRODUCTION OF PERSONAL COMPUTERS TO DATE IS AS FOLLOWS:, TRS-80 100,000 UNITS, PET 25,000 UNITS, APPLE 20,000 UNITS, NOT BAD CONSIDERING THAT APPLE STARTED SELLING AFTER PET, AND OFFERS A MORE EXPENSIVE SYSTEM. TALKING ABOUT OTHER MAKES, THE RUMOUR MILL IS WORKING OVERTIME ABOUT T.I. SEEMS, THAT A SYSTEM IS LIKELY TO BE ANNOUNCED BEFORE END OF YEAR IN THE $500-$1000 PRICE BRACKET. T.I. WILL PROBABLY ANNOUNCE A BUSINESS SYSTEM BY MID-YEAR SELLING FOR $5000 SYSTEM WILL HAVE A 500,000 BYTE ROM! ALSO 2 PRINTERS, AND DISKETTE. ROM WILL CONTAIN A VARIETY OF BUSINSS SOFTWARE.

MINI'APP'LES USER BANK, INCLUDED IN THE BANK ARE SEVERAL PROGRAMS SUBMITTED BY APPLE-PI OF COLORADO AS AN EXCHANGE FOR SOME PROGRAMS SUBMITTED ON OUR BEHALF BY KEITH MADONNA. WE ARE STILL WORKING ON A GOOD DISTRIBUTION PROCEDURE. UNTIL SUCH A PROCEDURE IS DEVISED AND PUBLISHED, PLEASE CONTACT EITHER K.MADONNA (474-3876) OR R. WENTWORTH (825-9086), IF YOU WISH TO FIND OUT HOW TO GET COPIES. A CATALOG OF THE BANK WILL BE PUBLISHED NEXT MONTH.

MINI'APP'LES
13516 GRAND AVENUE SOUTH
BURNSVILLE
MINN., 55337