

EXPANDABLE COMPUTER NEWS

Issue #22

September/October 1987

\$3.00

ECN

An Independent Publication for users of the Adam Computer
COPYRIGHT 1987 - SAGE ENTERPRISES

ADAM Suppliers.....	2
RAMBLINGS From the Ridge.....	3
Industry Observations.....	3
Bulletin Board.....	4
Expand Your Memory.....	4
Turbodisk Review.....	5
A Mouse for ADAM.....	6
More Design Features of FORTH.....	7
SmartBasic POKEs and PEEKs.....	8
User Groups.....	10
Shape Construction Program.....	13

ISSUE 22

COPYRIGHT 1987 SAGE ENTERPRISES

\$3.00

Expandable Computer News (ECN) is published bi-monthly by Sage Enterprises. Subscription rates are \$15.00 per year (6 issues) U.S. and Canada; \$21.00 foreign. All subscriptions payable to Sage Enterprises in U.S. funds only. Send all correspondence (subscriptions, ads, reviews, orders, articles and products) to: Sage Enterprises, Expandable Computer News, Rt. 2, Box 211, Scrivner Rd., Russellville, MO 65074.

Staff: Editor - Darrell R. Sage
Associate - Shirley I. Sage
Cover by D. Sage

NOTE: The views expressed by contributors to ECN are not necessarily those of the publisher. ECN and Sage Enterprises are not in any way affiliated with Coleco Industries, Inc.

Sage Enterprises accepts no liability for damage caused to any equipment pursuant to the construction of any hardware project.

We welcome contributions of original articles, programs, reviews, comments, etc. We are unable to pay for such contributions. We would like to thank all of our subscribers for their assistance—without them this publication would not be possible.

ADAM SUPPLIERS

Alpha-1, 1671 E. 16th Street, Suite 146, Brooklyn, NY 11229; ph. 718 336-7612.

M.W. Ruth Co., 3100 W. Chapel Ave., Cherry Hill, NJ 08002; ph. 609 667-2526.



Ramblings From The Ridge by D. Sage

We're back! Temporarily down, but not completely out. The failures we encountered that resulted in our last issue being real late have now been corrected. What happened? Glad you asked. After ECN was almost ready to go to the printer, I was doing some updates on our Zenith 100. Some of those take a while, so while one was running I got up and was doing something else to kill time. That's when I heard a loud pop. When I looked at the Zenith, I noticed the screen was blank. The monitor on light was still lit, but I didn't hear the fan on the computer running. The computer was dead with all lights out. I use the Zenith to write ECN and to maintain mailing lists and print labels. I keep all files backed up, but at this point the files were of no value since I did not have a computer to access them. Bad news. Even if I could access the files with another computer, they would be unusable because the files were stored in special formats by the data base and word processor I use. So the Zenith went to the repair shop.

The news was not good. The power supply was dead and it was likely that the rest of the boards were gone also. In order to check out the boards, I would have to shell out the \$300 necessary to replace it. The boards would cost more than the power supply. Not good at all. After considering my lack of options, I decided to go ahead and place my bets. The way the system died, I suspected that the problem was strictly in the power supply. As it turned out I was right. The Zenith is fixed and we were able to get the mailing labels printed.

All of this left me in somewhat of a dilemma. I could no longer be dependent on any one computer. Last spring, I finally bought an Atari 1040 ST. The Zenith uses 5 1/4 disks and the Atari uses 3 1/2 in. disks. I ended up buying a 5 1/4 in. drive for the Atari and decided to make ASCII copies of all files that I have on the Zenith. In theory, I should be able to transfer the ASCII files to the Atari. It isn't working yet, but its just a matter of time.

Anyway, this issue is being written on both the Zenith and the Atari. I use Word Perfect on the Zenith, which is the best word processor that I have ever used. It's supposed to be available for the Atari soon, but until it is I have been using 1ST Word and a desk top publishing program called Publishing Partner. They work fairly well together and give me access to different printer fonts, graphics, headlines, etc. Everything is now being printed on a Seikosha SP-1000A dot matrix printer which does a pretty good

job. The 1040ST came with a built in double-sided 720K disk drive and color monitor.

That and the printer cost me just over \$1100. The stand alone ADAM I use cost \$600 plus \$260 for one disk drive. The color monitor I use with ADAM cost another \$250 for a total of \$1125. Things sure are changing. I had more than that tied up in my first Commodore VIC-20 system.

You probably want to know - is he trying to make a point? Well, the point I am trying to make is that things have sure changed a lot since the home computer business got started. You can now buy a one megabyte system for the same price that you used to have to pay for a 16K system. Tomorrow and next year, and the next, who knows what you'll be able to get for around a \$1000.

My office is being remodeled. We are trying to install a bathroom and get things ready for winter. I can hardly wait. The summer was hot and dry and winter will probably be cold and wet.

The July 1987 issue of OMNI contains an article on telecommunications. In that article, Russ Walters (he wrote the first manual that Coleco included with ADAM) continued to criticize the ADAM.

After looking through the pile of stuff I have for this issue, I'm not real sure what will fit and what won't. I plan on including anything that didn't make the last issue as well as more programs, hardware projects and the regular stuff. Hope you enjoy it.

Amil Dillinger (ADAM User's Group #1986, 2226 Patterson Ave., S., Joplin, MO 64804-6322) has been busy locating a wide variety of ADAM products. Many of these products are extremely difficult to obtain from other sources. To get a catalog, send him a legal sized self-addressed stamped envelope.

Industry Observations by D. Sage

Commodore is finally shipping the new Amiga models in the U.S. I still have concerns that Commodore may have acted too late to save the company. The company is attempting to sell the Amiga 2000 in the corporate market place. Such a move will likely be unsuccessful. If Commodore is to regain the position that they achieved with the Commodore 64, then they will have to recognize that their future lies with the home market and not business markets. It takes more than PC compatibility to open the doors to corporate America.

I recognize that the press has beaten the PC compatibility issue to death, but having PC compatibility won't guarantee success. Corporate America is going to be more concerned with Commodore's past and current track record. Most large companies are unlikely to buy products from a

company that has until recently been in great financial difficulty. If Commodore doesn't get their act together before the Christmas selling season, we may find them begging Jack Tramiel (Atari Corp.) to bail them out.

Atari continues to sell computers. Both the 520 and 1040 are doing well. The multi-user operating system (OS9) is now available and the list of software continues to expand at a phenomenal rate, reminiscent of the Commodore 64.

IBM's share of the PC market continues to slip. Although the new PS2 series is doing well, the PC compatibles are doing better. Apple is also being quite successful with the new line of Macintosh computers.

The home market has been left entirely to Atari (XL & ST) and Commodore (64 & 128) in this country. No one else has been able to make serious inroads in the past and foreign companies appear to have lost any interest in the US market except in the area of PC compatibles.

Meanwhile, the video game market is thriving. Nintendo is leading the way with Sega following closely behind. Atari and Intellivision are bringing up the rear. In this area the American companies misjudged the market believing all the press hype that the home game market was dead.

Bulletin Board

The Bulletin Board is a free service to ECN subscribers. If you have something to sell, need to buy something, need help or just want to correspond with other ADAM owners, send a notice and ECN will publish it here.

■ **FOR SALE:** ADAM computer with disk drive and Magnavox 80 monitor. All manuals, Basic, AdamCalc, Packcopy, Smartfiler, MultiWrite, 20 disks & 20 tapes and more. \$350 for all. Contact: Ted Cherry, Rt. 1, Box 190, Cotter, Arkansas 72626, (501-430-5417).

WANTED: Complete solution to 2010: Text Adventure Game by Coleco. Also looking for a Vectrex Video Game System. Contact: John Bonavita, P.O. Box 320, St. Bonaventure, NY 14778.

■ **FOR SALE:** List #20 like-new ADAM software in all three formats (ROM, DDP, Disk) All at HUGE discount prices! Send a legal size, self-addressed stamped envelope for complete list. We do not sell pirated software! Contact: John Bonavita, P.O. Box 320, St. Bonaventure, NY 14778.

■ **WANTED:** Oh, no! I've been robbed! Insurance company says I must replace to receive reimbursement. I am looking for an Atari 5200 console, (I still have the switch box and power supply), any type of controllers (original or otherwise), and any games. I had nearly all that were available.

Please send list of anything you have (or can get) and your price list to Barbara Duncan, 3727-34th SW, Seattle, WA 98126.

■ **FOR SALE:** Eve Speech Synthesizer with software - \$75.00. Atari expansion unit - \$25.00 (prices include postage). Contact. Lee Smith, Box 159, Terre Hill, PA 17581.

■ **FOR SALE:** ADAM Expansion Module #3 w/Colecovision, tape drive, printer with on/off front switch and stand. \$200 worth of software. \$180 or best offer. Excellent condition. Contact George Nihem, 22609 Alger, St. Clair Shores, MI 48080; 313) 776-4755.

■ **FOR SALE:** ADAM Expansion Module #3, with disk drive, 64K memory expander, Eve SP-1 interface, and much software - \$500. Adam software: CP/M 2.2 with many public domain disks (\$25); SmartFiler (\$15); AdamCalc (\$15); Logo, Flash Card Maker, and more. Also carts - \$7.50 each: Smurf, Monkey Academy, Sir Lancelot, Zaxxon, and more. All software includes original disk or tape and manuals. Send SASE for complete list to Susan Lenz, 6210 West 117th Ave., Crown Point, IN 46307, 219/663-1805.

■ **NOTICE:** John Moore suggests that every CP/Mer get Tony Morehen's BIOS patch from CompuServe or another source. His new CCP is nice, but the patch that increases the M: drive to 61K and allows it to survive a computer reset will more than pay for itself!

Expand Your Memory by John Moore

While there is a good number of 64K memory expansion boards available, the dedicated "hardware hacker" can add extra memory to his or her ADAM for about \$10. Interested?

First, remember that any changes inside ADAM will void any warranty you may have left. Since we have no control over it neither the author nor ECN may be held responsible for the results of your work.

This modification requires good soldering skills, experience in working with circuit boards and with integrated circuits and a degree of patience.

First you will need the raw materials! 8 - 4164 dynamic RAM memory chips (200 nanoseconds a less), and about a foot of fine insulated wire. The chips are widely available for about \$1.50 each - check the Computer Shopper.

Open your ADAM and work your way to the bottom board. This is the one with the expansion connector and the three expansion slots on it. The memory chips are above and to the right of the connector marked "J5." This is Slot #3 - the place where your 64K Ram card would have been plugged.

The memory chips themselves will be numbered

in various ways, but will most likely have 4164 somewhere in there. On my board, the chips are labeled "TMS4164-20NL." The 20NL means 20 nanoseconds. Notice that the locator "notch" on the chips goes toward the back on the back row and toward the front on the front row (THIS IS VERY IMPORTANT)!

Now, prepare the expansion chips. Here is the way the little "bugs" are numbered. If you sit the IC on the table in front of you so that the "notch" points to your left, the bottom left pin is #1, the bottom right is #8, the top right is #9, and the top left is #16.

On each of your eight chips, carefully bend pin #15 up so that it is vertical (or nearly so). Do this carefully so that you don't break a pin. If you do, that chip is useless!

Now place one of the new 4164 chips on top of an existing chip. Be certain that the locator "notch" on both chips face the same way! Next, very carefully solder each pin on the new chip to the corresponding pin on the chip beneath it. There is a disc capacitor (C2) on the front row that may be troublesome. Carefully bend it out of the way as you solder.

You now have 8 new 4164 chips mounted "piggyback" on the old chips. Check your soldering very closely. You may well have created a solder "bridge" between pins. If you have, ADAM won't boot (but will otherwise be OK). Look carefully!

The final step is simple. Take the fine wire and connect each of the pin #15's of the new chips to the others. Finally, connect those pins to R15. This is a small resistor coded yellow-brown-black located between Slot #2 and Slot #3. There are two of them. You want the one nearer the back of the board. The wire connects to the side nearest Slot #3.

If you have completed these steps properly, you are now finished. Reassemble your ADAM, but don't put all the screws back in yet. Connect all the peripherals, and boot the machine. If SmartWRITER comes up, you are probably OK. If you have AdamCalc or BACKUP+ v. 3.0, start them running. Both will check the presence of expansion RAM.

If this test passes, you may consider the RAM functional. As one last check, boot CP/M and PIP something to Drive M:. The command PIP M:=A:PIP.COM[v] will copy PIP itself to the RAM Drive and will verify that an exact copy was made (the [v] switch). If the transfer goes OK - you're home free!

In my experience, the most likely trouble is the solder bridge between pins. Use no more solder than is necessary. If you get a bridge, the computer probably won't boot - not even SmartWriter. The good news is that this usually will not damage the chips. Once the bridge is cleared, everything will work fine.

If you don't have the necessary skills, check with a TV repairman or a Ham Radio operator in your area. The work is delicate, but not beyond the capabilities of the average technician. A side effect of this modification is that you don't need to add any extra suppression, since the busses on the main board have already been suppressed.

Product Review: Turbodisk

by D. Sage

Product: Ram Disk Utility

Manufacturer: Digital Express, Rt. 1, Box 29-G, Oak Hill, WV 25901

Media: Disk or DDP

Price: Review copy supplied by manufacturer

Turbodisk provides ADAM owners with a Ram Disk that is accessible in Basic. Also included with the software are utilities that allow you to turn your SmartKeys into special function keys in basic and a copy utility that uses the Ram Disk.

The instructions for the program come on two 8 1/2 x 11 in. sheets and are relatively easy to read and understand.

In order to use the program, you must first load SmartBasic, set Lomem and then Brun the Ram Disk



program. After it is loaded you can then reset Lomem to its original setting. In addition to creating a Ram Disk (D7) that is accessible in Basic, a number of Basic bugs are also fixed by the program. The fixes include the ability to recover "h" files and to INIT data packs and disks with the correct number of blocks in the directory (255 for data packs and 159 for disks). It also allows you to use Brun, Bload, and Bsave from a Basic program without the "Control Buffer Overflow" error message.

The Ram Disk program works fine. Of course you must remember that when ADAM is turned off,

everything in the RAM Disk will be lost. Ram Disks are especially helpful for temporary files and for storing programs that are used frequently.

The copy utility that comes with the program is fairly fast. It transferred 62 blocks using two data drives in about 2 1/2 minutes. The utility has a number of features that makes it fairly flexible. It uses the Smart Keys and has the labels for them on screen. My main complaint is that some of the labels were fairly difficult to read.

If you have the 64K Ram Expander and have a need for a Ram Disk in SmartBasic then you need this product. Recommended.

A Mouse For ADAM by Ron Smith

The following information will show you how to modify the Commodore 1350 Mouse for use with the ADAM. To use the mouse, you must first modify a Power Adapter for the Game Controller (see diagram).

PARTS: Plug-in-the-wall power supply providing at least 3.5 and no greater than 5 volts D.C. power output; 7 IN914 switching Diodes (Radio Shack Cat. #276-1620); two D-type mate connectors (Radio Shack Cat. # 276-1978); a small utility box for mounting the D type plugs; a soldering iron with a VERY SMALL TIP; a VERY SHARP KNIFE; and time and patience. **PLEASE NOTE:** You must connect the Positive (+) and the Negative (-) wires from the power adaptor to the correct locations in the Diagram. If they are reversed or connected to the incorrect pin locations, **DAMAGE TO THE COMPUTER** or connected device may occur. Also applying more than 5 volts DC current from the power supply may cause damage. As little as 3.5 volts can be used **BUT NOT MORE THAN 5 volts DC.**

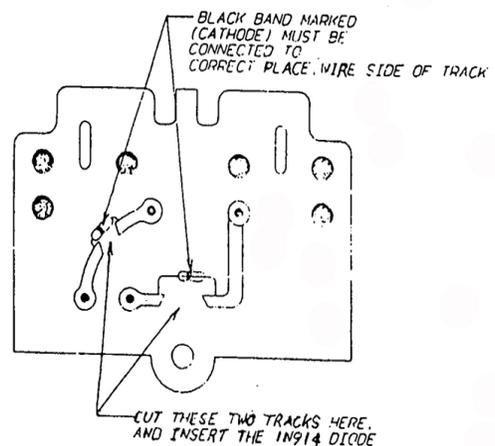
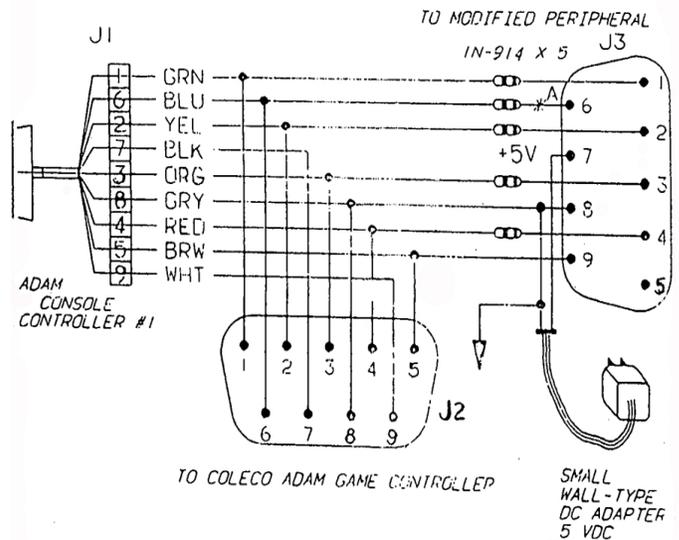
As the diagram indicates, you can leave the mouse and the regular ADAM joystick plugged into the connector once it is completed. J-2 on the diagram represents the connections for the regular game controller, while J-3 represents the connection for the mouse. The female end of the cable plugs into the computer and you build from there (after cutting the cable in half to shorten it). Save the other half for a possible future project. The sharp knife is needed to cut out holes in the plastic box for inserting the two D-type male plugs that will be mounted in the utility box, unless you can locate one that is pre-cut.

Once you have the power supply and the connecting cables correctly wired, its time to tackle the Commodore 1350 Mouse. Turn the mouse upside down and slide, according to the arrow markings, the cover mounting for the white ball. Remove the small piece of foam that protects the ball during shipping.

Remove the two screws near where the cable is connected and very carefully lift the cover. Don't force it, but sliding it slightly towards the cable connector may help. Next remove the screw that holds the small PC board in place. The board has two round pushbuttons on the top surface. Turn the board over and look at the circuit tracks on the underside. Use the sharp knife to cut and remove a section of the circuit track. Then solder a diode on each track as indicated in the diagram. Be careful to keep the solder within the circuit track area or you may ground the solder joint. A little dab will do yuh! Unsolder (desolder) the small blue and black wires and resolder them in reverse positions (the blue one where Commodore wants the black wire).

Now, put the PC board in place with the single screw. Replace the cover, then the white ball and its cover plate, and you're in business if you have done everything correctly.

The mouse can now be used with icon-driven software such as PAINTMASTER and PINBALL CONSTRUCTION SET. Good Luck.



More Design Features Of FORTH

by Thomas Gilmore

This is the fifth in a series of 5 articles on FORTH for the ADAM computer, now available in two public domain versions from EXPANDABLE COMPUTER NEWS.

The "Start-Up" version of FORTH for the ADAM is a set of files directly from FORTH-83, the 1983 International Standard. The first four articles focused on WHAT is available for your ADAM computer, how to put it to work, what FORTH is and isn't, and some of the specific FORTH advantages and design features.

In this article I will point out some of the other design features of FORTH and how they work, including a step-by-step discussion of another sample program. I will also get to the brief but happy story of how it came to be available for ADAM users. MINIMUM ERROR HANDLING

This is not a plus in everyone's eyes, especially at the beginning, but at least there are interesting trade-offs. You can write your own interrupts and error handling, as you need it, when you need it (particularly when debugging), and then remove what you don't need and get maximum speed. For ADAM owners starting out, it usually means a LOT more system crashes and "Computer Resets" than they might like.

NATURAL LIST PROCESSOR

The Forth word " " (pronounced "tick") allows a great variety of "pointers" used in AI programs. Forth is, in fact, a natural, list-processing language, usable for exploring computing problems which otherwise require a special language like LISP or PROLOG, neither of which, to my knowledge, is yet available for ADAM.

LEXICONS (VOCABULARIES)

Facilities are provided which allow, even encourage, the development of your (or a user's) special vocabulary for a "real world" problem. In other words, the user does not have to conform to pages and pages of a manual written for everyone's problem — a FORTH programmer, working with a user can make the computer talk the USER'S own language!! It takes a lot of learning, some experience, and (often) many tries, but it can be done.

"SHADOW" SCREENS

I have not taken the space to show any example of it in this series, but the Forth editor allows an easy pairing of a screen for documentation with another screen of code. Forth itself is completely documented in this way. All you need to do to go back and forth on-line is "A L" 'Alternate List'.

TRANSPORTABILITY

Particularly if you stay within the Forth-83 standard

vocabularies, all of your own words and programs can be quite sure of running on that bigger, faster machine you will move to some day.

Now, here's a short program for a "real world" business situation, although somewhat abbreviated and simplified for this article, to calculate, sub, and store a day's receipts (written later to disk or DDP):

```
5 list
Scr #5 G.BLK
0 \ticket-seller set-up and utilities 07Mar87tcg
1 \ Next line sets up the "tix" & "price" tables and
sets values
2 create tix 24 allot tix 24 erase create price 40 , 30
, 20 ,
3 \ Offsets for "tix" table no* set up for market & seat
price
4 0 constant MIL ( Milwaukee ) 8 constant MSP (
Minneapolis )
5 16 constant DMA ( Omaha )
6 0 constant $40 2 constant $30 4 constant $20
7 create city ( variable to hold addr start address in
"tix" )
8 : market ( S n -- d )
9 tix + city ! 0 s>d ;
10 : sold ( S d n2 n3 -- d )
11 2dup city @ + +! price + @ um*
12 3 spaces 2dup 7 d.r d+ ;
13 : summer ( S d -- d ) 12 spaces 8 0 do ascii - emit
loop ;
14 : add ( S d -- )
15 summer cr 16 spaces 7 d.r ;
ok
```

Lines 0, 1, and 3 are completely comment lines. Line 2 and lines 4 thru 7 set up a 3 by 4 storage table, as if it had columns for cities and rows for each seat category. 24 bytes are reserved, since we need TWO bytes per data cell. ["n" represents a default, 16-bit number on the data stack; "d" represents a double-number, or 32-bit size.]

Lines 8 and 9 define the first word for our user. It takes the market name (abbreviation) that he provides, converts it to a numerical offset, finds the address of the "top of the column", stores it in the variable "city", and sets up a double-size zero on the data stack.

Lines 10 thru 12 do most of the real work: duplicates are made of both the number sold and the category, the number sold is added to what's already in the "tix." table (0, in this version), the dollars are calculated, dup'd; printed out, and added to the total ["d"].

Lines 13 thru 15 define the last 2 words to show the total dollars for a city.

And here's what it looks like when you compile it, run it, and check the results in the updated "tix" table:

```
5 load ok
MIL market ok
600 $40 sold 24000 ok
300 $30 sold 9000 ok
200 $20 sold 4000 ok
add -----
          37000 ok
MSP market ok
500 $40 sold 20000 ok
400 $30 sold 12000 ok
100 $20 sold 2000 ok
add -----
          34000 ok
DMA market ok
70 $40 sold 2800 ok
80 $30 sold 2400 ok
90 $20 sold 1800 ok
add -----
          7000 ok
```

The hex dump reveals that the two bytes are stored in reverse order:

```
tix 24 dump
0 1 2 3 4 5 6 7 8 9 A B C D E F
5F60 7B 01 58 02 2C 01 C8 00 00 00 F4 01 90 01 64
00
5F70 00 00 46 00 50 00 5A 00 00 00 ..
decimal 600 hex . 258 ok [matches OK with "5802"]
decimal 300 hex . 12C ok [matches OK with "2C01"]
decimal 80 hex . 50 ok [matches OK with "5000"]
decimal 90 hex . 5A decimal ok [and with "5A00"]
```

This public domain version of Forth-83 was made possible thru the original authors, Henry Laxen and Mike Perry of Berkeley, CA, the providing of a Z100 version by Fred Olson, leader of the Minnesota Forth Interest Group, and the CONVERT program of Sage Enterprises.

This is the last article of this series on FORTH. I've enjoyed bringing it to you, and I hope some of you have by now immersed yourself in the new, wide-open world of FORTH on your ADAM computer. Enjoy!

SmartBASIC Peeks & Pokes Version 3 (Revised 8/12/86) by John Moore

There are a number of useful POKES and PEEKS available to the SmartBASIC programmer, but since they have never been collected into one work, many of the more recent ADAMites have been "locked out"

of their use. This article is an effort to collect as many of these as possible so that the knowledge can be shared.

CAUTION: The locations given here are known to be correct for version 79 of SmartBASIC. This is the most widely distributed version of the program. To find out which version you have, load SmartBASIC and type "PRINT PEEK(260)'return'." The number you see is the version number. If it's less than 79, check with another member of the ADAM Support Group about getting the most recent version.

1146 - This is where ADAM stores his "prompt" character (usually "]") which appears at the start of every line. POKEing any ASCII value here will display that as the new prompt.

6356 - To modify the LOAD command so that the old program will not be erased when you LOAD a new program, poke a 201 into this location. This will let you merge new programs with old programs in memory by LOADING them. **WARNING!** Be sure to reset this to the default of 205 when you are finished merging. Any old line numbers that are used by the new program will be overwritten!

12185 - This location controls the length of text lines that ADAM will permit. You may increase this to 239 characters. Long program lines are an aid in translating certain Applesoft BASIC programs.

16089 - This location (and 16090) contain the current HIHEM setting.

16095 - This location (and 16096) holds the value of LOMEM.

16126 - This location (and following) give the line number in which an error occurred. An error handling routine can obtain information with a PEEK.

16129 - This is the SPEED storage. Normally it is 255. You can POKE a new value here instead of using the SPEED command.

16134 - This contains a 3 - the ASCII value of Control-C. This is the only way SmartBASIC has of telling what this key is. To disable the BREAK function, just POKE some value here that does not have a keycode (like 255). POKE the 3 back to turn the BREAK back on!

16149 - This location (and 16150) control the highest address that SmartBASIC will let you POKE. Normally 16149 is 144 and 16150 is 211. If you change both of these to 255, you will be able to POKE to any address in ADAM's memory!

16178 - This is the number of significant places given in a floating point number. The default is 9. This **ONLY** affects output. ADAM will use as many digits as possible internally.

16641 - This is the indicator of the boot (start-up) device. An 8 indicates Tape #1, 4 is Disc #1, etc. Programs can look here to see what device started

everything running.

16763 - This is the "X" value of the H PLOT command.

16764 - The "Y" value of the H PLOT command.

16765 - Holds the value of the SCALE command.

16779 - Value of PDL(6) "left trigger": 1 - On, 0 - Off.

16780 - Value of PDL(8) "right trigger" (as above).

16781 - Value of PDL(12) "keypad number pressed." The value of the number is returned here with "*" = 10, "#" = 11, and nothing pressed = 15.

16783 - This is PDL(5) "direction" 1 - Up, 2 - Right, 4 - Down, 8 - Left.

16784 - PDL(7) "left trigger" #2 controller (see 16779).

16785 - PDL(9) "right trigger" #2 controller (see 16780).

16786 - PDL(13) "keypad # - #2 controller (see 16781).

16788 - PDL(4) "direction - #2 controller (see 16783).

16821 - Here is the location where ADAM stores the device number of the drive currently in use. This normally changes when you give a command (such as CATALOG,d1), but you can POKE the new drive here. The Device ID's are 8 (tape 1-D1), 24 (tape 2-D2), 4 (disk 1-D5) and 5 (disk 2-D6).

16953 - This location contains the character being used by ADAM for the cursor. The default is 95 - the underline mark. To turn off the cursor, you can POKE a 0 here.

16954 - ADAM does not really "erase" a screen when you command HOME. It actually fills the screen with spaces. The ASCII code for a space is 32, which is the default value of this location. If you POKE some other value here and command HOME, it will fill the screen with that character. This could be useful for special effects or title screens.

17000 - POKE in a 1 here will stop the cursor from flashing, but the character above the cursor will not be displayed. You can reset this by POKEing a 0 here or using the TEXT com`zmand. See 17291.

17001 - Cursor row location.

17002 - Cursor column location.

17059 - This contains the screen color code. The screen color is also called the "border," since it is always visible around the edges of the ADAM screen. You must command "TEXT" after changing this value before any effect can be seen.

17115 - This contains the color controls for the normal text screen. It is calculated according to the example given later. You must use the TEXT command to see the result.

17126 - This contains the color controls for the

reversed video characters - the ones seen when FLASH and INVERSE commands are used. The calculation is given later. TEXT must be executed before you see any effect.

17164 - Any ASCII character POKEd here will fill the screen when the command TEXT is used. Compares with 16954 for the HOME command. Useful for graphics or special effects - particularly when paired with 16954!

17198 - This is the number of lines on your screen. The default is 23. This is another command which requires the use of TEXT before any result is apparent.

17199 - This is the number of columns on your screen. The default is 30. If you change this, you must use TEXT.

17201 - The location of the Top Margin on your screen. ADAM's default is 0.

17202 - This is the location of your left margin. The default is 1 (first column). You can use the last two locations to create a smaller area or "text window" for use with some custom designed graphics.

17291 - Controls the cursor blinking rate. 0 is steady on with 255 the slowest flash rate. Stopping the cursor from flashing by POKEing a 0 here is preferred to using the location at 17000.

17302 - This location contains 16 - the ASCII value of Control-P. You can disable this feature by POKEing this location with a number that does not have a keycode (like 255). Poke a 16 to turn the Control-P feature back on!

17954 - This location controls the pitch of the tone you hear when you PRINT CHR\$(7). The possible values range from 0 (highest possible tone) to 63 (lowest note).

17963 - is the storage for the variable that determines the duration of the tone you hear when you PRINT CHR\$(7). Values are permitted from 0 - 255. The larger the number, the longer the "beep"!

18607 - Determines the screen (border) color in GR mode. It will take effect after a "GR" command.

18633 - Contains the color of the GR window back-ground after the "GR" command.

18711 - Is the color value of the text in the window at the bottom of the screen in GR mode. It is determined in the same way as either normal or inverse text. It takes effect after the "GR" command.

25431 - This is the screen (border) color in HGR mode. Takes effect after an "HGR" command.

25471 - Is the color of the graphics window in HGR mode. Takes effect after "HGR" command.

25558 - Contains the value for text displayed at the bottom of the HGR screen. You determine this value as with all other text color values. Only takes

effect after the "HGR" command.

27100 - This is the input value from Paddle 0 (0-255).

27101 - Here is the input value from Paddle 1.

27002 - And Paddle 2.

27003 - With this the input value from Paddle 3.

64885 - This is the location of ADAM's "last character typed" buffer. This is very useful if you have a program that you want to let the user interrupt. Each time through a loop, PEEK at this location. If the value you want is returned, arrange for your program to take proper action!

CALCULATING COLOR VALUES

ADAM's color palette has the following colors, and set of associated values.

0 - Transparent	8 - Medium red
1 - Black	9 - Light red
2 - Medium green	10 - Dark yellow
3 - Light green	11 - Light yellow
4 - Dark blue	12 - Dark green
5 - Light blue	13 - Magenta (pink)
6 - Dark red	14 - Gray
7 - Cyan (bluish)	15 - White

To set the screen value, simply poke the color value into the specified location. The "color" transparent is intended to be used in either the main or inverse values to let the screen color show through the characters.

To create the color value to POKE into the main or inverse screen locations, you must do one small calculation (per value). The number to be POKEd is the sum of 16 x (character color value) and the background color value.

As an example, black letters on a dark yellow background would be 10 (dark yellow) + 16 x 1 (black) = 26.

Normally, you would not want to make the background and the text the same color, but sometimes you may want to "blank" the screen while sending output formatted for the printer. The easiest way to do this is to make the characters and screen the same color while printing, then switch back when you're through.

Some locations listed here (like 16126) are actually two locations. This is because the variable is so large that one byte isn't enough. The two locations are called the HIGH byte and the LOW byte.

To convert to a value (like a line number) that makes sense to you, multiply the HIGH byte by 256 and add it to the LOW byte. As an example, if an error occurred and through PEEKing, you found that 16126 was 10 and 16127 was 201, then the error

occurred at line (201*256)+10 or 51466! The low byte is always given first.

If you know of other locations for PEEKs and POKEs that you are willing to share, please pass them along to:

The ADAM SUPPORT GROUP

c/o John Moore
1970 Fisher Tr., NE
Atlanta, GA 30345

User Group News

The list of users' groups continues to grow. If there isn't one in your area to join - start one!

NATIONAL GROUPS

#1 Adam User's Group
P.O. Box 3761 - Attn: Jay Forman
Cherry Hill, NJ 08034

LOCAL GROUPS

James E. Gilbert
4608 Lakeview Dr.
Huntsville, AL 35810

Victor L. Watford
P.O. Box 777
Russellville, AL 35653

Richard Bains
7210 Bulen Drive
Anchorage, AK 99507

Danny Levitt
4525 S. White Pine
Tucson, AZ 85730

Robert R. Marentes
9425 N. 38th Ave.
Phoenix, AZ 85021

East Bay ADAM Group (EBAG)
Tom Ozretich
6097 Slopview Court
Castro Valley, CA 94552
ph. 415-886-2884

Harvey Klein
So. California ADAM Users
1736 So. Bedford Street
Los Angeles, CA 90035

Frank Fleisch
13381-19 Magnolia Ave.
Corona, CA 91719

Central Calif. Adam User's Group - James Turner, Jr.
20110 Ave. 19
Madera, CA 93637

San Diego Adam Users Group
Dr. Harold Alexander
37 Catspaw Cape
Coronado, CA 92118

AUG of San Diego County
868 N. 2nd St. #242
El Cajon, CA 92021
ph. 619/445-2400

Bay Region ADAM Information Network
550 27th St. #202
San Francisco, CA 94131
ph. 415/282-3056

Inland Empire Users Group - Ann Quetel
6644 Seine Ave.
Highland, CA 92346
ph. 714/862-5807

Denver ADAM User's Group
1416 Lipan St.
Denver, CO 80204

ADAM Users Group #305 - John F. Busby, II
6634 SW 41st St.
Davie, FL 33314

Emerald Coast ADAM User's Group
Howard Pines
1010 Gloria Drive
Ft. Walton Beach, FL 32548

Robert J. Niemeyer
292 Boca Ciega Point Blvd. N.
St. Petersburg, FL 33708

ADAM User's Group - Michael G. Graham
217 Albert St.
Winter Springs, FL 32709
ph. 305/327-1387

ADAM Support Group - John Moore
1870 Fisher Tr. NE
Atlanta, GA 30345

Hawaii AUG
Harlan Fletcher
2335C Apollo Ave.
Honolulu, HI 96818
ph. 808/422-4019

Donald R. Lager
5415 N. 2nd St.
Rockford, IL 61111
ph. 815/877-7786

KC Users Group
Joe Reardon
1513 Tauromee
Kansas City, KS 66102
913/371-7491

Greater Cincinnati Adam Users Group
c/o Keith Bowman
P.O. Box 434
Alexandria, KY 41001

ADAM Network
P.O. Box 85
East Detroit, MI 48021

Bill & Nancy Rahn
12426-15th St. S.
Afton, MN 55001
ph. 612/436-6577

Outsider's Users Group
Donald Viltiard
P.O. Box 771
Starkville, MS 39759

Omaha ADAM Users Club
Norman Castro
809 West 33rd Ave.
Bellevue, NE 68005
ph. 402/291-4405

Al Roginski
4327 Thorndale Pl.
Las Vegas, NV 89103

Metro Adam User's Group
Russell Williams
414 W. 149th St.
New York, NY 10031
ph. 212/208-0645 (9am-5pm M-F)

Genesee Valley Adam Users
Donald K. Zimmermah
5132 Jordon Road
Silver Springs, NY 14550

ADAM-X-Change (New York & Canada)
Wade Rowley
12863 Washburn
Wolcott, NY 14590

Tri-Angle Adam Users
L-5 Oak Grove
Chapel Hill, NC 27514
ph 919/968-0299

Mutual ADAM Users Group
412 Bettie Street
Akron, OH 44306

Lake Erie Adam Users
2110 W. 36th Street
Lorain, OH 44503
ph. 216/282-8467

Portland Adam Users Group
Craig Frerichs
P.O. Box 1081
Portland, OR 97207

The (717) Adam Users
Steve Chamberlain
120 E. 4th ST.
Bloomsburg, PA 17815

Midsouth ADAM Users
Roger Burford, Lot 142 NAS MHP
Millington, TN 38053

Adam Users of El Paso
Alan Samuels
4821 Vista Del Monte
El Paso, TX 79922

Houston AAUG
c/o Thomas Rutan
1805 14th Ave. N
Texas City, TX 77590

Norfolk ADAM Group - Gerald M. Steen
1000 Rockbridge Ave. #144
Norfolk, VA 23508

ADAM Users Group of Central Virginia
Thomas J. Kelly
3B, Rt. 664
Earlsville, VA 22936

ADAM Washington D.C. Users Group
1811 St. Roman Dr.
Vienna, VA 22180

Puget Sound Adam Network
22607 SE 322nd
Kent, WA 98042
ph. 206/886-1167

Dave Sandahl
USNH, Box 2844
FPO Seattle, WA 98778

CANADA

Robert Dunstan
95 Harland Crescent
Ajax, Ontario L1S 1K2

Derek Townsend
Box 820
Claresholm, Alberta T0L 0T0

Edmonton Adam Users Group - Richard Forges
14712 - 122 St.
Edmonton, Alberta T5X 1V9

J.A. Girard
1420 Ave. Langevin Sud
Alma, Quebec G8B 6B1

Mike Laurier
7350 Roi Rene
Anjou, Quebec H1K 3G6

Mr. G. Hibbert
P.O. Box 10
Mistatim, Saskatchewan S0E 1B0

First Canadian Adam User's Group
P.O. Box 547 Victoria Station
Westmount, Quebec H3Z 2Y6

Winnipeg Adam Users Group
729 Government Ave.
Winnipeg, Manitoba R2K 1X5

Metro-Toronto Adam Group
P.O. Box 123
260 Adelaide St. East
Toronto, Ontario M5A 1N0

AUSTRALIA

The Bendigo Colecovision Club
C1-2 Fenton St.
Bendigo, VIC 3550, Australia

ADAM Owner's & User's Group
4 Norman Street
Deakin, ACT 2600, Australia

ENGLAND

The U.K. ADAM Subscribers - Keith A. Marner
33 Homer Road
Croydon, Surrey, CR0 7SB, England

High Resolution Shape Constructor

by David Clark

```
100 REM High Resolution Shape Constructor
110 REM
120 REM
130 REM Created by David Clark
140 REM 2/12/85
150 REM
160 REM
170 REM
180 REM
190 x = PEEK(16095): y = PEEK(16096): POKE
26580, x: POKE 26581, y
200 lo = y*256+x+3001: LOMEM :lo
210 x = 127: y = 95: ze = 0: an = 1: th = 3: tn = 10:
true = 1: false = 0: ia =
PEEK(26581)*256+PEEK(26580)
220 DIM m%(3000), u%(3000)
230 POKE 17115, 240: POKE 17059, 4: TEXT
240 fi = true: s = INT(ia/256): POKE 16766,
ia-s*256: POKE 16767, s: POKE ia, 0
250 d$ = CHR$(4): GOTO 290
260 s = PEEK(ia): IF s = 0 THEN RETURN
270 ia = PEEK(ia + s*2)+PEEK(ia+s*2+1)*256:
FOR i = ia+ia TO ia+ia+200: IF PEEK(i)=0
THEN i = i-ia: RETURN
280 NEXT: RETURN
290 TEXT: GOSUB 260: z$ = STR$(i) : byt$ = "<
+z$+" BYTES >"
300 PRINT "SHAPE TABLE START ADDRESS IS
AT": PRINT TAB(INT(31-9)/2); "< "; ia;" >"
310 VTAB 5: PRINT "THERE ARE "; PEEK(ia); "
SHAPES IN MEMORY"
320 VTAB 8: PRINT " SHAPE TABLE TOTAL
LENGTH": VTAB 10: HTAB
INT(31-LEN(byt$))/2: PRINT byt$
330 VTAB 14: PRINT "1 - CREATE NEW SHAPE":
PRINT: PRINT "2 - SAVE TO DATA PACK":
PRINT: PRINT "3 - LOAD FROM DATA PACK":
PRINT
340 PRINT "4 - SAVE TO PRINTER": PRINT:
PRINT "5 - END";
350 GET w$: IF w$ < "1" OR w$ > "5" THEN 350
360 ON VAL(w$) GOTO 370, 390, 440, 1140, 450
370 l=0: t = 0: u = 0: mm = 0: x = FRE(8): x = 127:
y = 95
380 POKE 25431, 4: POKE 25471, 1: HGR:
HCOLOR = 3: d = 1: GOTO 590
390 s = PEEK(ia): IF s = 0 THEN 290
400 ia = PEEK(ia+s*2)+PEEK(ia+s*2+l)*256: FOR
i = ia+ia TO ia+ia+200: IF PEEK(i) = 0 THEN i
= i-ia: GOTO 420
410 NEXT: GOTO 290
420 GOSUB 990: PRINT: PRINT d$; "BSAVE P,A";
ia; ",L"; i+1-ia
```

```
SHAPE TABLE START ADDRESS IS AT
< 27407 >
THERE ARE 0 SHAPES IN MEMORY

SHAPE TABLE TOTAL LENGTH
< 0 BYTES >

1 - CREATE NEW SHAPE
2 - SAVE TO DATA PACK
3 - LOAD FROM DATA PACK
4 - SAVE TO PRINTER
5 - END_
```

```
430 PRINT d$; "CLOSE P" : PRINT d$; "RENAME
P,"; fi$: GOTO 290
440 GOSUB 990: PRINT: PRINT d$; "BLOAD "; fi$;
", A"; ia: fi = false: GOTO 290
450 POKE 17115, 23: POKE 17059, 4: TEXT: a =
PEEK(26581)*256+PEEK(26580): LOMEM :a:
END
460 POKE 25431, 4: POKE 25471, 1: HGR:
HCOLOR = 3
470 GOSUB 1090
480 IF VAL(z$)+mm = 3000 THEN 1420
490 IF mm = 3000 THEN 1420
500 HTAB 10: VTAB 24: PRINT "BYTES="; u+1; :
IF w$ <> CHR$(160) THEN 520
510 f = 0: GOSUB 1010: y = y-1: IF y<0 THEN y=0:
GOTO 470
520 IF w$ <> CHR$(162) THEN 540
530 f = 2: GOSUB 1010: y = y+1: IF y>191 THEN
y=191: GOTO 470
540 IF w$ <> CHR$(163) THEN 560
550 f = 3: GOSUB 1010: x = x-1: IF x < 0 THEN x =
0: GOTO 470
560 IF w$ <> CHR$(161) THEN 580
570 f = 1: GOSUB 1010: x = x+1: IF x>255 THEN
x=255: GOTO 470
580 IF w$ <> CHR$(129) THEN 630
590 k = k+tn: IF d = 1 THEN HOME: d = 2: PRINT
" SMARTKEY I SMARTKEY VI";: PRINT:
PRINT: PRINT " DRAW " ;:GOTO 610
600 GOTO 620
610 PRINT " MORE"; : GOTO 470
620 HOME: d = 1: PRINT " SMARTKEY I
SMARTKEY VI";: PRINT:PRINT " DON'T
DRAW", " MORE"; : GOTO 470
630 IF w$ = CHR$(134) THEN r% = r%+1: GOTO
1330
640 IF w$ = CHR$(130) AND r% = 1 THEN 660
650 GOTO 670
```

```

660 u%(u) = t: GOTO 290
670 IF w$ = CHR$(131) AND r% = 2 THEN 690
680 GOTO 790
690 POKE 25431, 4: POKE 25471, 1: HGR:
HCOLOR = 3: x = 127: y = 95
700 FOR i = 0 TO mm-1
710 w = m%(i): IF w >= 4 THEN w = w-4: HPLOT x,
y
720 ON w+1 GOTO 730, 740, 750, 760
730 y = y-1: GOTO 770
740 x = x+1: GOTO 770
750 y = y+1: GOTO 770
760 x = x-1
770 NEXT
780 GOTO 1330
790 IF w$ = CHR$(132) AND r% = 3 THEN 810
800 GOTO 470
810 u%(u) = t: n = u: POKE 17115, 240: POKE
17059, 4: TEXT
820 IF u%(0) = 0 THEN 290
830 VTAB 13: HTAB 11: INVERSE: PRINT
"PROCESSING": NORMAL
840 IF fi THEN fi = false: GOTO 860
850 GOTO 890
860 POKE ia, 1: POKE ia + 1, 0: POKE ia+2, 4:
POKE ia+3, 0
870 FOR i = 0 TO n: POKE ia+4+i, u%(i): NEXT
880 POKE ia+4+i, 0: GOTO 290
890 ns = PEEK(ia): la = ia+ns*2 ls =
ia+PEEK(la)+PEEK(la+1)*256
900 i = ls
910 IF PEEK(i) = 0 THEN 930
920 i = i+1: GOTO 910
930 FOR j = i+2 TO la+2 STEP -1: POKE j,
PEEK(j-2): NEXT
940 na = i-ia+3: ms = INT(na/256): ls = na-ms*256:
POKE la+2, ls: POKE la+3, ms
950 FOR j = 2 TO la-ia STEP 2: ls = PEEK(ia+j): ms
= PEEK(ia+j+1): ls=ls+2: IF ls>= 256 THEN
ms=ms+1: ls = ls-256
960 POKE ia+j, ls: POKE ia+j+1, ms: NEXT
970 FOR j = 0 TO n: POKE ia+na+j, u%(j): NEXT:
POKE ia+na+j, 0
980 POKE ia, ns+1: GOTO 290
990 PRINT:PRINT: INPUT " FILENAME: "; fl$: IF
fl$ = " " THEN POP: GOTO 290
1000 RETURN
1010 g = (d-l)*4 + f: m%(mm) = g: mm = mm+1
1020 HCOLOR = (d-1)*3: HPLOT x, y: k = 5
1030 IF l < 2 THEN 1060
1040 IF g > 0 AND g < 4 THEN t = t+8^l*g: GOTO
1080
1050 u%(u) = t: u = u+1: t = g: l=l: RETURN
1060 IF l = 1 AND g = 0 THEN u%(u) = t+192: u =
u+1: t = l: l = 1: RETURN
1070 t = t+8^l*g: IF l < 2 THEN l = l+1: RETURN
1080 u%(u) = t: u = u+1: t = 0: l=0: RETURN
1090 GET w$: IF VAL(w$) < 128 THEN 1110
1100 w$ = CHR$< 5 THEN RETURN
1110 k=k+1: IF k<5 THEN RETURN
1120 k = 0: IF c = 0 THEN c = 1: HCOLOR = th:
HPLOT x, y: RETURN
1130 c = 0: HCOLOR = ze: HPLOT x, y: RETURN
1140 s = PEEK(ia): IF s = 0 THEN HOME: VTAB 10:
PRINT CHR$(7);: PRINT " THERE IS NO
SHAPE IN MEMORY";: GOTO 1160
1150 HOME: VTAB 4: PRINT " PLACE PAPER IN
THE PRINTER PRESS ANY KEY WHEN
READY "; : GET w$: VTAB 8: GOTO 1180
1160 PRINT CHR$(7): PRINT: FOR i = 1 TO 1000:
NEXT i: PRINT " PLEASE PUT A SHAPE IN
MEMORY": FOR i = 1 TO 1000: NEXT i
1170 PRINT: PRINT TAB(8); "!! THANK YOU !!":
FOR i = 1 TO 3000: NEXT i: GOTO 290
1180 la = PEEK+PEEK(ia + s*2+1)*256: FOR i =
la+ia TO la+ia+200: IF PEEK(i) = 0 THEN l =
i-ia: GOTO 1200
1190 NEXT i: GOTO 290
1200 PR #1: line% = 10000: bl = i
1210 s = INT(ia/256): ss = (ia)-s*256
1220 PRINT line%; " LOMEM "; ia+l+1
1230 PRINT line%+10; " POKE 16766, "; ss; ".
POKE 16767, "; s; ": ad = "; ia
1240 PRINT line%+20; " FOR i = 0 TO "; l; ": READ
d: POKE ad+i, d: NEXT i": llne% = line%+30
1250 FOR i = 0 TO 1 STEP 16: PRINT line%; "
DATA ";
1260 FOR t = 0 TO 15: IF t > 0 THEN PRINT ", ";
1270 IF ia+i+t = bl THEN PRINT "O": GOTO 1320
1280 PRINT PEEK(ia+i+t);
1290 NEXT t: PRINT: line% = line%+10
1300 IF line% = 10550 OR line% = 11110 OR line%
= 11670 THEN PR #0: GOSUB 1550
1310 NEXT i
1320 PRINT CHR$(13): PR #0: GOTO 290
1330 IF r% >= 4 AND d = 2 THEN r% = 0: d = 1:
GOTO 590
1340 IF r% >= 4 AND d = 1 THEN r% = 0: GOTO
620
1350 ON r% GOTO 1360, 1380, 1400
1360 HOME: PRINT " SMARTKKY II", "SMARTKEY
VI": PRINT: PRINT " DEL SHAPE &", " MORE"
1370 PRINT " GOTO MENU"; : GOTO 470
1380 HOME: PRINT " SMARTKEY III", "SMARTKEY
VI": PRINT: PRINT " RESTORE ", "MORE"
1390 PRINT " SHAPE"; : GOTO 470
1400 HOME: PRINT " SMARTKEY IV",
"SMARTKEY VI": PRINT: PRINT " PUT SHAPE
", "MORE"
1410 PRINT " IN MEMORY"; : GOTO 470

```

```

1420 HOME: FOR i = 1 TO 8: PRINT CHR$(7):
      NEXT: HOME: PRINT "THIS SHAPE", "
      SMARTKEY V":
      PRINT "CAN BE NO", : PRINT
1430 PRINT " LARGER", "DEL (OR) SAVE": PRINT
      , " LAST SHAPE";
1440 GET w$: IF w$ <> CHR$(133) THEN 1440
1450 HOME: PRINT " SMARTKEY II", " SMARTKEY
      IV": PRINT: PRINT "   DELETE", "SAVE":
      PRINT " LAST SHAPE". " LAST SHAPE";
1460 GET w$
1470 IF w$ = CHR$(130) THEN r% = 1: GOTO 670
1480 IF w$ = CHR$(132) THEN r% = 3: GOTO 790
1490 GOTO 1460
1500 HOME: FOR i = 1 TO 8: PRINT CHR$(7):
      NEXT: HOME: PRINT "THIS SHAPE", "
      SMARTKEY V": PRINT "TABLE IS", : PRINT
1510 PRINT " FULL ", "DEL SAVE": PRINT , "
      LAST SHAPE";
1520 GET w$: IF w$ <> CHR$(133) THEN 1520
1530 HOME: PRINT " SMARTKEY II", " SMARTKEY
      IV": PRINT: PRINT "   DELETE", "SAVE":
      PRINT " LAST SHAPE", " LAST SHAPE";
1540 GOTO 1460
1550 HOME: VTAB 10: HTAB (31-19)/2: FLASH:
      PRINT "PLEASE CHANGE PAPER": NORMAL
1560 VTAB 23: PRINT " PRESS ANY KEY TO
      CONTINUE"; : GET w$
1570 HOME: PR #1: RETURN

```