Volume 2 Issue 8 $1.75 JANUARY 1987

"Serving 99'ers Since 1984"

THE SMART PROGRAMMER

Editor's Message

Many of you have written in support of the newsletter and to inquire about its status. Several months ago, I undertook writing several assembly routines to speed the tasks involved in publishing the newsletter and, ironically, the project has evolved into a course away from timely publication. Once the project was underway, it became obvious that the routines had potential for being useful to other users. While it was a difficult choice, I elected to pursue distribution, a decision that was strongly influenced by the fact that many recent products for 99'ers have been hardware-specific, while the routines I was working on could contribute to filling the void in software releases designed for a broader spectrum of users. Incorporation of the user interfaces required to transform the routines into a solid package took considerable time, longer than I anticipated. The project is now basically completed (see page 5), so I have now returned my attention to publishing the newsletter. I hope, as I'm sure you do, that issues will be released on a more frequent basis. This publication has always been dedicated to quality, original material and that often requires considerable development time, so some delays are almost inevitable, but I will endeavor to produce issues as often as possible. Many thanks to everyone who has been supportive. Yes, the newsletter will continue. The purveyors of gloom and doom rhetoric are once again wrong.

Richard M. Mitchell, Editor

Spring Shows

by Richard M. Mitchell

Major shows highlighted the spring activities of 99'ers. Reports indicate that The New England Fayuh (Boston), TICOFF (New Jersey), TI-Fest (Ottawa) and Fest-West (Los Angeles) were all well attended and the organizational efforts of the individuals and groups behind the shows were certainly appreciated by many users.

My fingers were pried from the keyboard long enough for me to catch a flight to L.A., so below is a report on my impressions of the products featured at Fest-West.

First, I want to thank all the folks in California who made my visit enjoyable. Terrie Masters, Tom Freeman and George Steffen helped me a lot personally, but more importantly they are totally devoted in their support of the 4A community every day of the year. Also, thanks to all of the user groups that participated at Fest-West: LA 99er UG, Brea 99 UG, SF Valley UG, Oxnard 99 UG and Tri Valley UG. I was also especially pleased to have the opportunity to visit with staff writers Mariusz Stanczak, Doug Warren and Craig Miller (Barry Traver was at the show in Ottawa that weekend).

Vendor addresses can be found at the end of the article.

MYARC 964Ø Computer

First, I want to point out that my only opportunity to date to view the 964Ø running was at Fest-West. What I saw was, overall, quite impressive. The hardware seems to be functioning properly, in accordance with its design. As I anticipated with a new hardware offering, most of the new software was still under development (several major items were near completion), though many 4A programs were functioning properly.

Perhaps the two most impressive features of the 964Ø are its overall speed and its video resolution. The 964Ø's TMS9995 CPU is fully instruction compatible with the TMS99ØØ used in the 4A and is faster. The 9938 video processor can utilize 7 modes of graphics operation and is also software compatible with the 4A's TMS9918A. The video output is crisp and the various modes allow implementing 32, 4Ø and 8Ø-column displays, as well as true bit-mapped graphics.

MYARC has placed considerable emphasis on compatibility with the 99/4A. While some trade-offs were made so as to provide support for new features, the 964Ø is compatible with many 4A programs. A new key scanning routine, designed to take advantage of the 964Ø's keyboard, was implemented in lieu of the 4A's direct key access capabilities, so that programs that accessed the 4A's TMS99Ø1 for key accesses are partially or completely incompatible with the 964Ø. Most software from TI did not access the TMS99Ø1 for keystrokes and many user-written programs could likely be modified, but a number of distributed software packages will have to be re-written by the authors to be usable.

As with the 99/4A, the PC/XT and many other computers, the 964Ø's direct memory addressing is limited to 64K. However, MYARC has implemented a memory paging system that makes accessing memory beyond 64K far easier than on the 4A. The 964Ø's 512K memory can also be configured with 64K as simulated GROM and 16K as simulated cartridge ROM, so that a 4A mode can be implemented. Overall, I'm extremely impressed with the paging system. However, I do hope that MYARC has taken into account memory mapping for Terminate and Stay Resident (TSR) programs, as I anticipate that quite a few controversies could arise over programmers' usage of memory. While I have seen no info on such mapping, it may already exist. But, such problems pale in comparison with attempts to manipulate memory on the 4A! I believe one could count on one hand all of the 4A programs that access pages of memory!

MYARC plans to soon be releasing its disk operating system, MDOS, approximately 88K of code that will enable many capabilities, including programming in native mode. Commercial native mode software will likely be available in time for the Fall show circuit. Public domain and fairware offerings are anticipated on the communications networks shortly after release of MDOS because preliminary versions of MDOS are available on networks such as GEnie and Compuserve for software development.

At the time of Fest-West, about 6Ø 964Ø's had been shipped. 1Ø were sold at Fest-West, a sell-out of the available inventory, by L & M Systems (Les Merriman), the MYARC West Coast Distributor. Many folks were clearly impressed. Reports indicate that the total number of units sold by MYARC is now well into the hundreds, at a typical retail price of about $55Ø including the 964Ø card and a keyboard (monitor, PEB, drives and PEB cards not included). My congratulations to MYARC -- producing the 964Ø is an incredible feat for any firm and especially one as small as MYARC. It's been a long time coming, let's hope it's a long time here!

Print Designer

Pesaca Soft is marketing a program called Print Designer that provides enhanced density printing on Micronics SG-1Ø and NX-1Ø printers. The program can be used in conjunction with GRAPHX, TI-Artist or BIGCHAR (Pesaca's own graphics design program, included with Print Designer). The superiority of Print Designer output over standard output from the same printer has to be seen to be believed! Fest-West was really buzzing about the quality of the program's output! The suggested price for Print Designer is $29.9Ø, an appropriate price for quality specialty software (the author is not going to get rich spending many hours developing a program for 99'ers with Micronics printers).

Pesaca also showed Charamat, a graphics development tool for XB programmers. Charamat is written in Assembly and operates quickly and smoothly. Charamat is priced at $39.9Ø.

Triton Turbo XT

MG was showing their 99/4A and PC products at Fest-West and also handled displays of the products they were contracted to implement for Triton (Turbo XT and Super Extended BASIC). Since I've already discussed the features of the Turbo XT previously, I'll just cover what I've since experienced.

I ordered a Turbo XT shortly after it was announced and have been quite impressed as far as an XT machine goes. While I continue to use the 99/4A for most tasks, I do find the XT to be quite useful at times. And, for those who are wondering, yes, I am using a 99/4A keyboard and a shared monitor and I find the saving of desk space to far outweigh the minor inconveniences of such an arrangement (my desk frequently looks like a war zone anyway, so I don't need anything extra on it!), though I realize such an arrangement will not suit everyone's preferences. I do find that the XT is greatly enhanced by a good monitor. For anyone interested in monitors, with the variety of hardware coming along, I recommend the purchase of a monitor that can display composite and both TTL and analog RGB -- you may not need all of those capabilities at present, but it is generally a good investment for the future. Some monitors, such as some models from Thomson and Magnavox also have green (or other color) screen switches for text intensive work.

MG was also promoting their VID program, which runs on PC's. VID is a video tape database program and has been reviewed as a product that meets the high standards of previous MG products.

XB:Bug

XB:Bug won last year's Computer Shopper 4A Programming Award and has received excellent reviews. The program, written by J. Peter Hoddie and released by Genial Computerware, is, as its name implies, a debugging tool for Extended BASIC programmers. The program is accessed with a simple keystroke or through a CALL LINK.

XB:Bug is a tremendously powerful tool that is amazingly easy to use for anyone with at least minimal experience in XB programming. It works with all current versions of XB that are similar to TI XB, including TI, GK, Mechatronics and Triton versions. In other words, CALL VERSION (X) should return below 200. MYARC XB II is a 128K implementation of BASIC that is completely different internally from TI XB, so XB:BUG will not work with it.

XB:Bug allows taking a look at virtually all of the items of interest to XB programmers -- variable values; character, color and sprite information; DATA/READ status; file buffer information; you name it!

There really aren't any programs even similar to XB:Bug and it is light years ahead of our XB programming tools of the past! It won't write a program for you, but it is the next best thing.

Gram Packer 1.1

Peter Hoddie has updated his excellent Gram Packer program. Included is more efficient file handling and an improved manual. We've previously covered the original program, so suffice it to say that this one is a must for anyone with a GRAM device! Gram Packer is available from Genial Computerware.

Wordwriter

DataBioTics displayed a new word processing cartridge called Wordwriter. The package has appeal for a wide range of users, as it can be used with or without memory expansion (12K of text without memory, 36K with memory) and supports use of a hard disk, including a Show Directory that will catalog a full path name. There is even a version, Wordwriter +, with its own printer interface and cable extending directly from the cartridge, so a user can be up and running with no expansion box!

Wordwriter follows many of the conventions of the TI-Writer Editor and sports several enhancements, including saving files in several different formats. Unfortunately, the cartridge cannot be dumped with a Gram Kracker and files saved with a Save File option do not carry a TI-Writer Tab record (DTB may revamp the SF to include the Tab record), but the package is otherwise quite impressive.

Wordwriter was written by Todd Kaplan, a talented programmer from Chicago. The suggested retail price is $39.95.

Wordwriter + (includes parallel printer interface) is $49.95. The parallel printer cable is $19.95.

DataBioTics plans to release at least a dozen (!) new products by this Fall, so look to DataBioTics to be an important part of the continued success of the 99/4A community.

Fairware

Fairware was championed by Steve Mehr, who did an outstanding job, selling at a minimal price a booklet listing over 2ØØ Fairware offerings and demonstrating many of the programs over the course of the show weekend. Ray Kazmer and Fairware author Ken Gilliland shared the honors in running the booth. There was a lot of interest in music, game and database programs.

Dijit AVDP Card

Dijit demonstrated their latest development, an Advanced Video Processor Card. The card uses a Yamaha V-9938 video processor mounted on a peripheral expansion card to provide many video capabilities, including 8Ø column display. Perhaps the most exciting portion of the firm's flyer for the product is mention of a Digitize Function, which allows a TV picture to be instantly adapted to a computer graphics picture! Dijit is currently working with several programmers to provide full software support for the AVPC.

Dijit also offers their earlier RGB conversion kit, a Cache Card (Dijit's "supercart") and several other products.

Intern

T.A.P.E. offered Heiner Martin's book Intern. Intern includes a disassembly of console ROM and GROM and is quite useful. The disassemblies run straight-through, so caution must be exercised in determining whether sections of code in a particular console might be offset from the book's listing, but the commented code in the book is certainly a bargain at the $12 that I paid.

Rave 99 Keyboards

Rave flew in from Connecticut to offer their popular IBM-style keyboards for the 4A and reportedly sold quite a few units. Rave has also introduced a card that allows a speech synthesizer to be placed in the expansion box, which is a matter of convenience on the 4A and necessity if you want speech on the 964Ø. Rave is continuing their R&D efforts, so look for more fine products from the firm.

Dots-Perfect

Myra White offered the Dots-Perfect upgrade for Epson FX, JX, RX and MX printers. The Dots-Perfect PROM is easily installed and provides dramatically improved print quality.

DATASYSTEMS

DATASYSTEMS' George Holod offers 99'ers a variety of specialty software (Chemtutor, Geography, Calculus, etc.) that many of you may find quite useful -- the sort of items you won't find available from any other vendor. Be sure to write to DATASYSTEMS for complete information!

Vendor Addresses:

MYARC, Inc., PO Box 14Ø, Basking Ridge, NJ Ø792Ø-1Ø14
Pesaca Soft, Box 828, Mendocino, CA 9546Ø
MG, 1475 W. Cypress Ave., San Dimas, CA 91773
Genial Computerware, PO Box 183, Grafton, MA Ø1519
Databiotics, PO Box 1194, Palos Verdes, CA 9Ø274
Dijit Systems, 4345 Hortensia St., San Diego, CA 921Ø3
T.A.P.E., Ltd., 1439 Solano Place, Ontario, CA 91764
Rave 99, 112 Rambling Road, Vernon, CT Ø6Ø66
Myra D. White, 849 E. Bonita Ave., La Verne, CA 9175Ø
Software Systems, 23Ø1 Churchill Drive, Oxnard, CA 93Ø33.

SHOW SCHEDULE

Chicago TI-Faire, November 7, 1987, Ironwood Room, Triton College, River Grove, IL.
Milwaukee Fair, November 8, 1987, Airport Quality Inn, 5311 So. Howell Ave., Oak Creek, WI
Fest-West '88, February 27 and 28, 1988, Palace Station Hotel and Casino, Las Vegas, NV

TI-99/4A

ALL THE NEW & EXCITING DataBioTics CARTRIDGES

FOR THE TI-99/4A COMPUTER

AT SUBSTANTIAL DISCOUNTS!

16K WORD PROCESSORS, MULTI-SCREEN GAMES, SPREADSHEETS, PRINTER INTERFACES, ALL BRAND NEW AND AT LOW PRICES!

NEVER BEFORE!! New double excitement in cartridge from the thrill masters who wrote Parsec, Munchman, TI-Invaders and Car Wars!!

BARRAGE

The fiendish Draks are raining terrible acid-balls on the planet. With skill, strategy, fun and action! you control awesome firepower . . . This laser cannon environment is controlled by one or two players. . . . . **$16.95 + S & H**

SPOT SHOT

No other game like it! You are a giant dragonfly. Your laser-like tongue and agility are more than a match for attacking insects . . . but you must fly over and around hedges to stay alive. . . . . **$16.95 + S & H**

BLACK HOLE

Listen . . . If you liked Parsec, you'll love Black Hole! Dynamic special effects put you in control of two spacecraft against the Black Hole Empire. One or two players. (Joysticks required.) . . . . **$16.95 + S & H**

Four Star Classic

JUNKMAN JR.

The year . . . 2487 A.D. The place . . . Aging planet Earth in the dying city of Burrwin, Illinois. You are collecting pollution for transport to an outer space junkyard. But you must listen for vicious micro-dogs. Only your agility will save you!

Entertains with lively music, spritely action, imaginative graphics and sound. From DataBioTics. (Joysticks recommended.) . . . . **$16.95 + S & H**

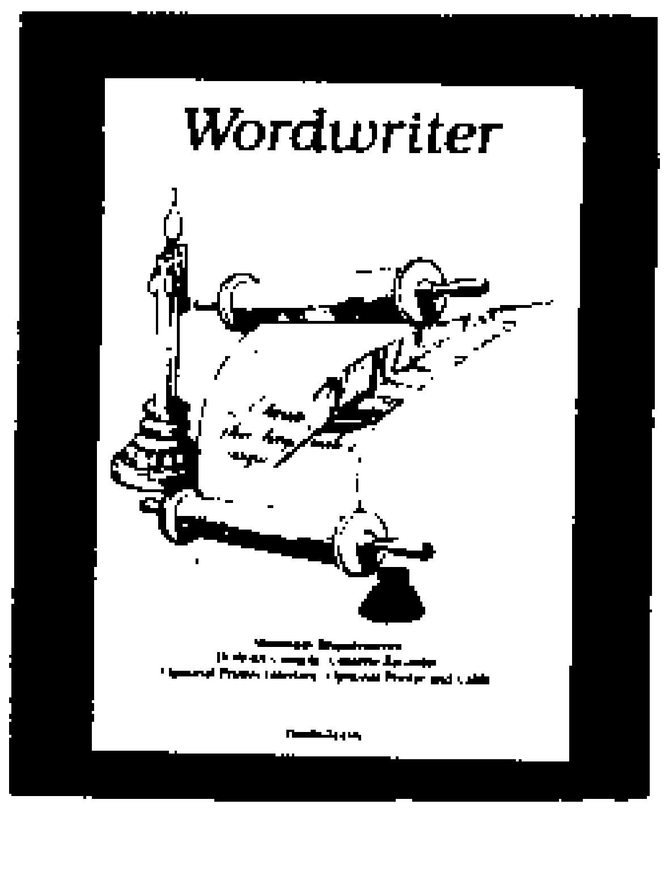
Science Fiction Thriller

STAR RUNNER

You are a captive, aboard the starship Arcturus, laden with plundered treasure taken from your planet.

Your mission . . . recover the stolen booty stored on all 25 decks of the starship, one deck at a time. Your only weapons are blinding speed and your blaster.

Music, sound effects and colorful visual cues combine to make *Star Runner* the best entertainment value ever for your 99/4A. From DataBioTics. (Joysticks recommended.) . . . . **$16.95 + S & H**



Magic In a Module

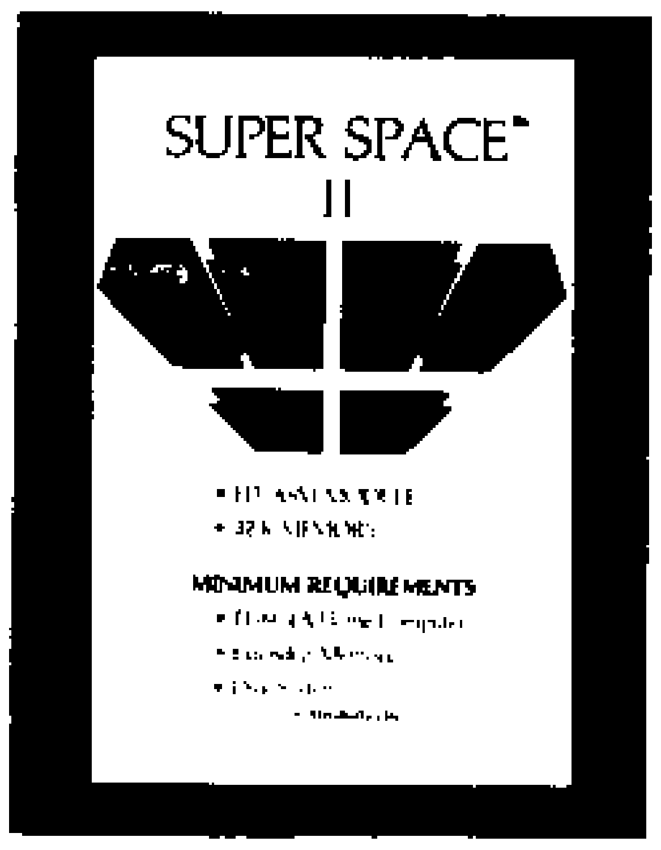
WORDWRITER

Simply plug your *Wordwriter* cartridge into the module slot and instantly you can: create mounds of text, up to 12,286 characters (36,782 with memory expansion); insert and delete characters, lines or whole paragraphs easily; turn work wrap on and just type; turn line number display on or off at will; reformat text to neat margins after changes are made to your document; search and easily find key words or phrases within your document; and set left and right margins and horizontal tabs where you want them from the keyboard.

Wordwriter allows you to load and save from disk or cassette, as well as print a hard copy. From DataBioTics. (Requires RS232 Interface and printer cable or *Parallax-TI Interface* and Printer.) . . . . **$34.95 + S & H**

WORDWRITER+

With *Wordwriter+* you need only your console and a printer. Includes all the features of *Wordwriter* (above) PLUS it has a 20-pin edge connector for direct hookup to your printer. Printer cable included. (Requires Parallel Printer.) . . . . **$54.95 + S & H**



Not Just Another Memory Expansion

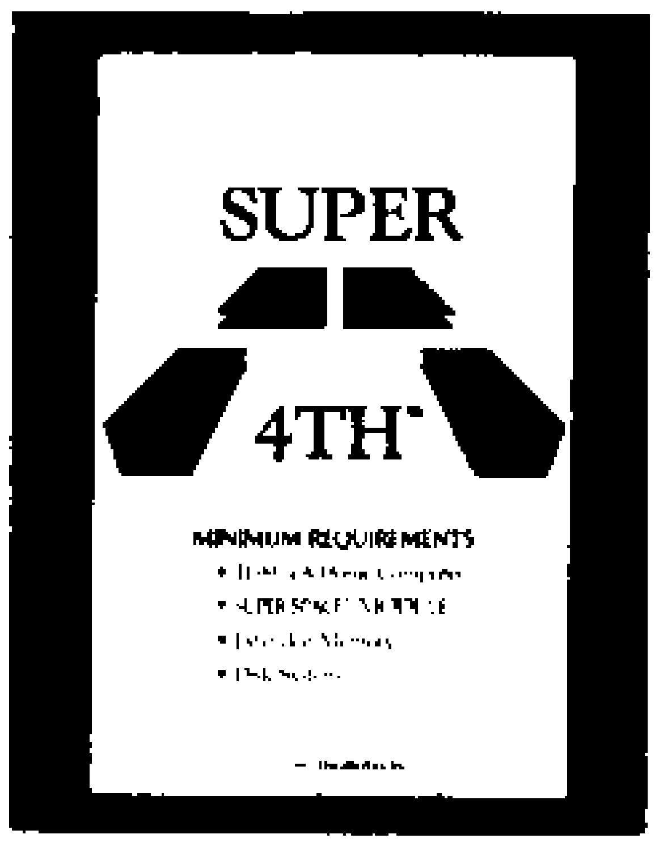
THE SUPERSPACE II 32K MEMORY MODULE

Have you ever wished for a cartridge with all your favorite software? *SuperSpace II* gives you the power to create your own cartridge-based software. Load in your own assembly code, vacuum software out of other cartridges, or even save your favorite ROM cartridges to disk and run them at the touch of a single key! Design your own menu loader for up to 7 Assembly Language programs with this ingenious module, equipped with a battery to save your programs in memory for at least two years.

Included with the cartridge in this great package is a built-in Editor/Assembler GROM and a free Editor, Macro Assembler disks, and the *SuperSpace II* manual, plus *Introduction to Assembly Language* and another bonus book. To get you started, a menu and cartridge loader are provided, along with E/A utilities, GROM header templates, a bitmap graphics demo and a free "C" compiler. (Requires Disk System and Memory Expansion. . . . . **$59.95 + S & H**

THE SUPERSPACE 8K MEMORY MODULE

All the features of *SuperSpace II* but with 8K of memory. Does not include the books or "C" compiler. . . . . **$34.95 + S & H**



SUPER 4TH

Almost as fast as Assembly, but easier to learn. *Super 4th* supports speech and sound, multiple disk drives (including Hard Disk!). Comes with Decompiler and 109 page manual. Requires *Superspace I or II* and disk system. . . . . **$17.95 + S & H**

Super Spreadsheet

CONSOLE CALC

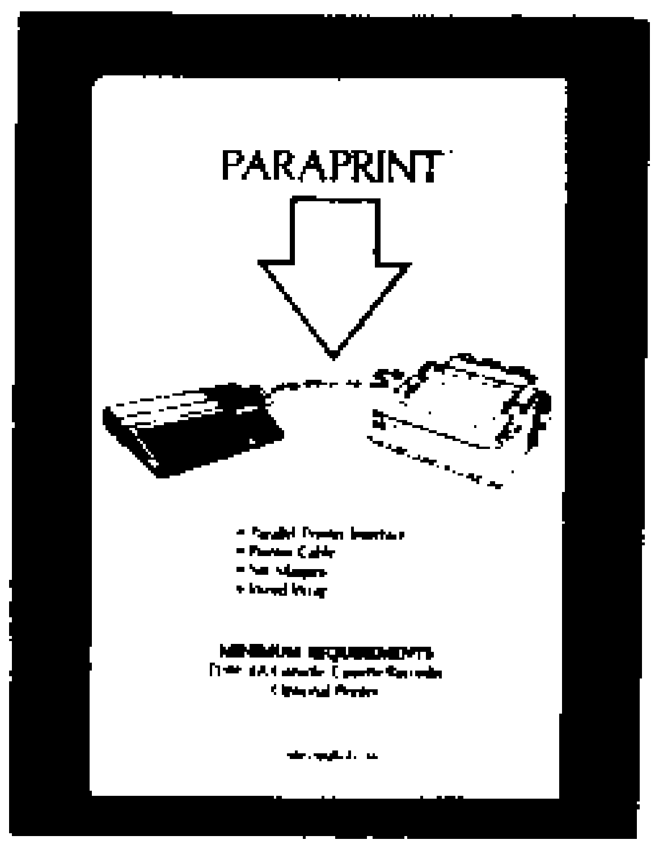
Console Calc is a new electronic spreadsheet built into a cartridge for convenience. Its calculator-like simplicity makes it a joy to use. Best of all, because it's a cartridge, all you need is your 99/4A.

Console Calc allows you to track income and expenses and prepare income tax records easily. Constructing your own professional-looking forms, table sand charts for business or school using the 11 simple commands is even a snap with *Console Calc.*

Console Calc expands its usefulness as you expand your system. A 864 cell spreadsheet enlarges to 2520 cells with *32K Memory Expansion.* Load and save your records to cassette or disk; or get a hard copy on your printer if you have one. Written in 100% Asembly Language from DataBioTics . . . . **$24.95 + S & H**

CONSOLE CALC+

Same as above with the addition of a built-in parallel printer interface which lets you plug any standard parallel printer directly into *Console Calc* with the supplied cable. Start printing your work without the need to purchase any other equipment. (Requires parallel printer.) . . . . . . **$44.95 + S & H**



PARAPRINT PRINTING CONNECTION

New from DataBioTics, the most advanced parallel printer interface for the TI-99/4A computer. With *Paraprint*, you can connect any parallel-input printer directly to your compter . . . simply plug one end into the side of your console, and the other end into your printer.

Paraprint does not split lines ever like the others. *Paraprint* lets you set both left and right margins. It even lets you turn word wrap on and off from the keyboard!

There is even a special word wrap that you control when listing your BASIC or EXTENDED BASIC programs. . . . . **$39.95 + S & H**

Name ______________________________
Address ______________________________
City ______________ State ______________ Zip ________

QTY.	ORDER NUMBER	ITEM	TOTAL
		Sub Total	
		Calif. Res. Add 6½%	
		Shipping & Handling	1 50
		Total	

Send order and make checks payable to:
DATABIOTICS, INC
P.O. Box 1194, Palos Verdes Estates, Ca. 90274

VISA

master charge

TEX-COMP
VISA and MASTERCARD HOLDERS CALL DIRECT
(818) 366-6631
24 Hour Order Line

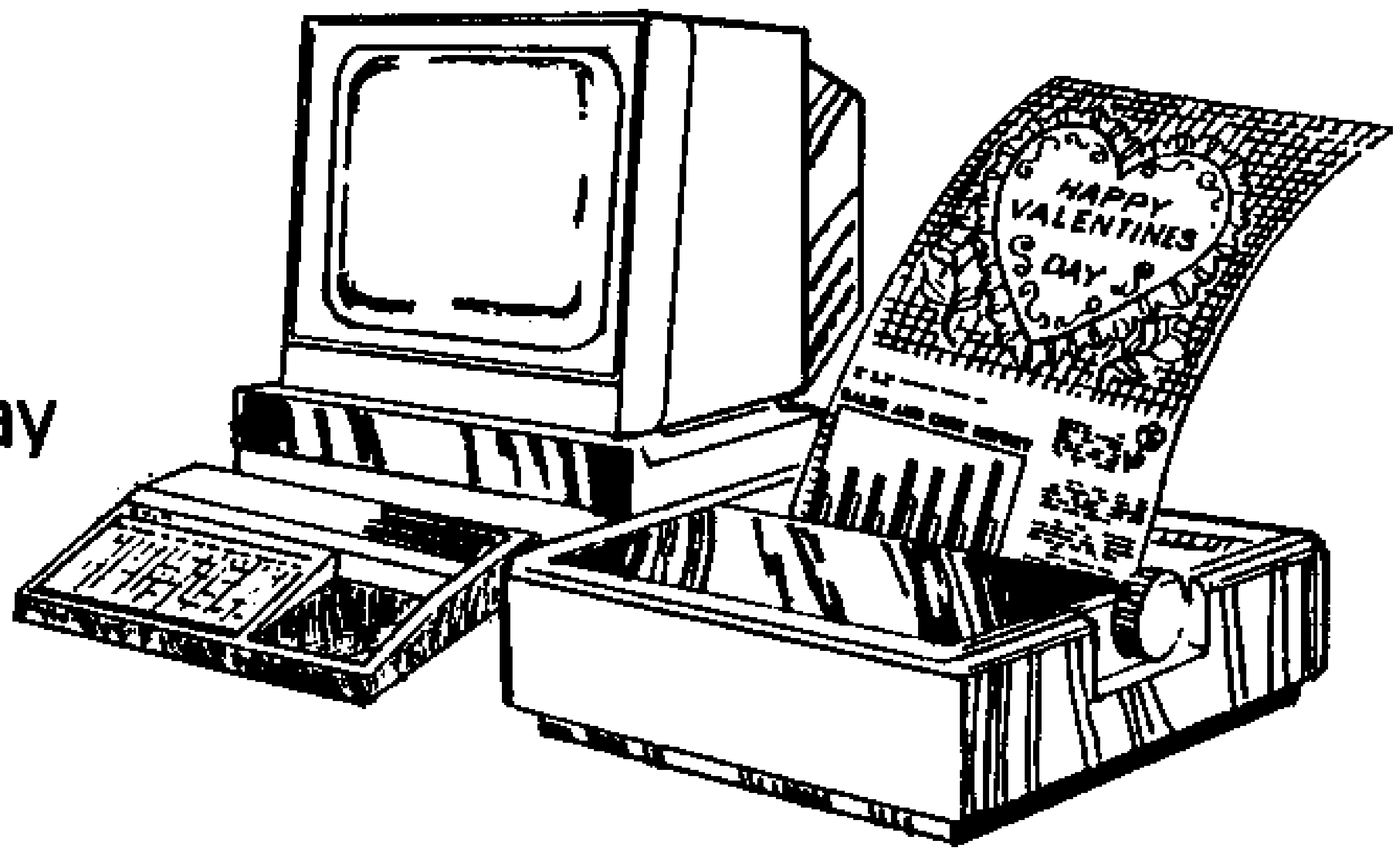
PHONE SERVICE

NOTE: Payment in full must accompany all orders—credit card, company check or money order for immediate shipment. Personal checks require up to 4 weeks to clear. California orders add 6½% sales tax.

REQUIRES ONLY YOUR CONSOLE!

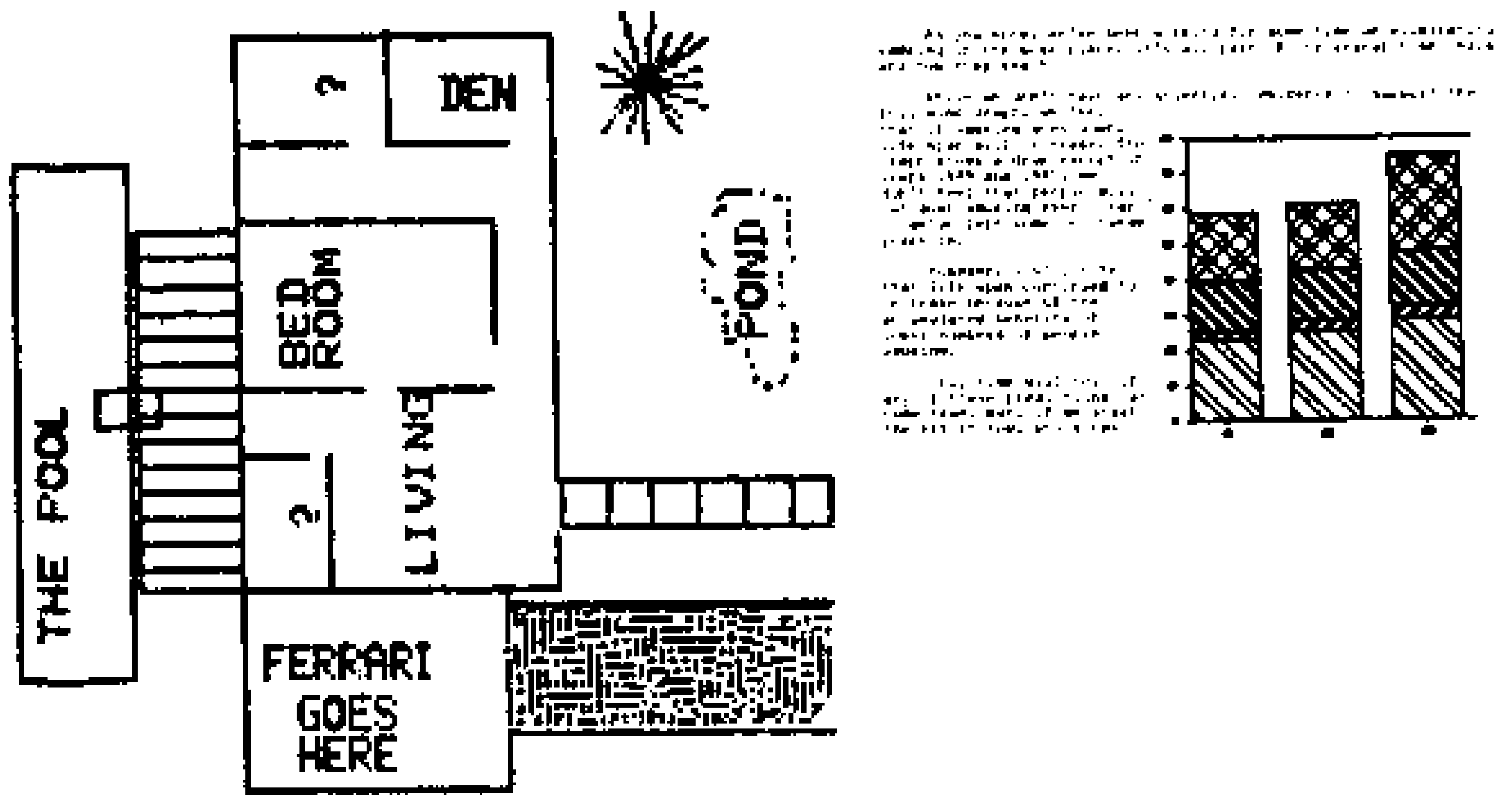
DESKTOP PUBLISHING MIRACLE

FOR YOUR TI 99/4A



It's a breakthrough that needs only your console and a printer. The hottest new way to use your 99/4A! Now you can compose greeting cards, graphic letters, impressive school reports, striking invitations, even if you can't draw a line!

IT'S EASY AND IT'S IMPRESSIVE



You get a word processor that creates beautiful fonts and a picture editor that selects from supplied changeable pictures. You can even draw your own! It's not complicated to use and you get stunning results.

PROJECT YOUR IMAGE

With **DESKTOP PUBLISHER**, you've written your last blah letter. Pick a type font to match your mood! Select a picture to emphasize a point. Doodle with line art to customize your signature. The possibilities are limitless.

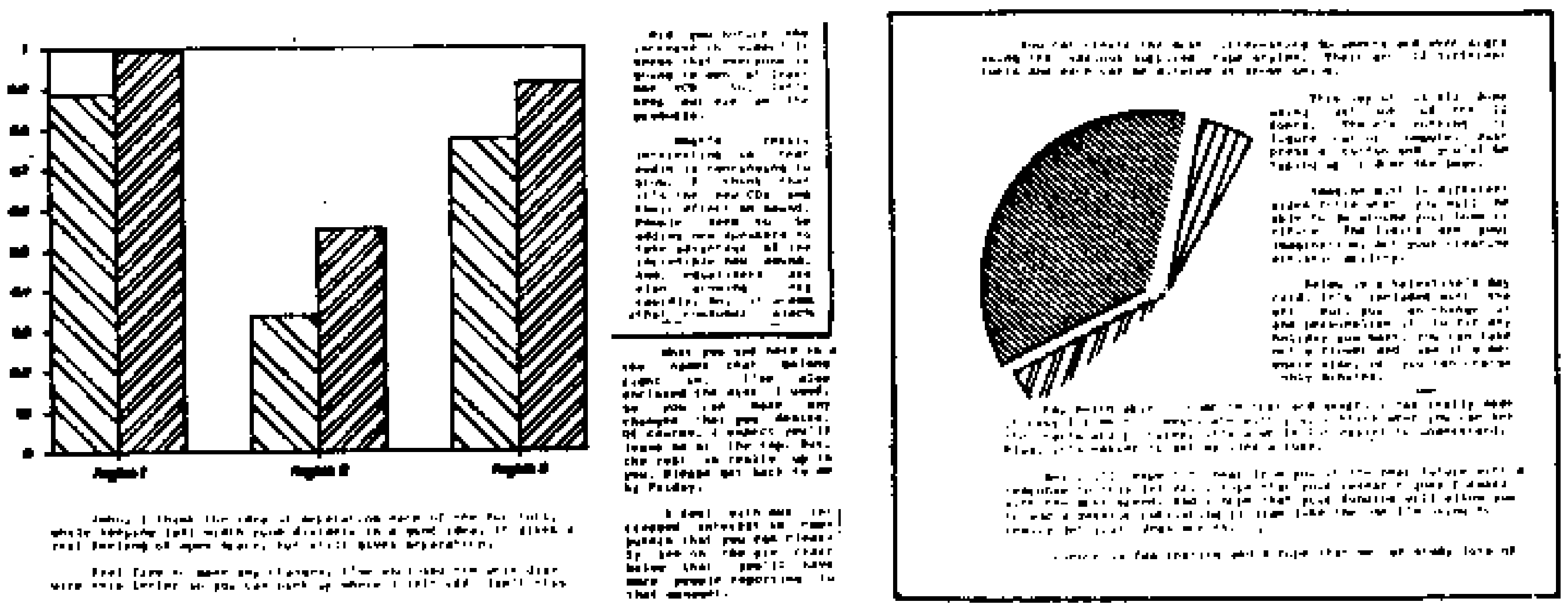
USE WITH ANY PRINTER

Use **DESKTOP PUBLISHER** with any standard Centronics parallel printer and interface; all you need to publish your own invitations, newsletters, menus, or start your own publishing business. You'll have powerful looking documents no matter what printer is used, because the secret is integrated text and graphics.

```
******************************
* SHIPPING SEPTEMBER 1987 *
******************************
```

AND YOU GET PRODUCTION RIGHTS

Save your own designs on cassette or disk. Birthday or special occasion? Load your card design back from cassette or disk, edit to fit the occasion and print it, all in minutes. You can design and save that church bulletin, advertisement, or report in neat 2 or 3 column pages, complete with illustrations and charts. Looks expensive. Looks complicated, but it isn't!



ALL YOU NEED TO START

Imagine leveraging the capabilities of your 99/4A console and your printer to produce the documents you see here. There's nothing else to buy. And not even dealers have **DESKTOP PUBLISHER** yet. Send your check for $59.95 (**$1.50** postage & handling). Order No. DBTDPM. California residents add sales tax.

Name ______________________________
Address ______________________________
City/State/Zip ______________________________

QTY	ORDER NUMBER	ITEM	TOTAL

Subtotal ________
Calif. Res. add 6½% ________
Shipping & Handling 1.50
TOTAL ________

Send order and make checks payable to:
DATABIOTICS, INC.
P.O. Box 1194, Palos Verdes Estates, CA 90274

String Master Alpha Testing Completed

by Richard M. Mitchell

Alpha testing has now been completed for an upcoming Bytemaster software release, String Master. The package is a set of Assembly Language routines, accessible from Extended BASIC, that provide macro equivalents of XB functions, thereby greatly reducing program development time and program run time. String Master's speed and practical design provide an environment that is conducive to creative programming.

String Master's conventions are based on TI standards:

All routines can be fully utilized with any standard 99/4A system with disk, memory expansion and Extended BASIC.

TI Extended BASIC and direct derivations therefrom are fully supported (MYARC XB is not supported because it is internally radically different).

Both Option Base Ø and Option Base 1 are fully supported.

Any string type (direct string, string variable, string array element or string array) is valid for any string parameter.

Any numeric type (direct number, numeric variable, numeric array element or numeric array) is valid for any numeric parameter.

Both string and numeric arrays can utilize any number of dimensions supported by XB, 1 to 7.

Complete error checking with standard XB error messages is provided.

Numeric integers within a maximum range of -32768 to 32767, inclusive, are fully supported.

String Master is useful for database applications, program development, text handling, screen windowing and many other applications. Routines are designed to provide optimization of usage of system memory. Over 28 routines are included in the package, with most designed for handling string arrays. Unlike the 255-byte limitation of traditional string handling through string variables, many String Master functions allow string arrays to provide access to string data with the only limitation being available memory.

String Master is $19.95, plus $2.ØØ domestic postage and handling ($6.ØØ foreign). Complete printed documentation with programming examples will be included. It is anticipated that the package will ship in late August.

Irresistible Resistor Project

An Eight Cent Hardware Mod

by Richard M. Mitchell

Being very software oriented, over the years I've left most of the hardware fun to others. Recently, I came upon an article by Bob Lawson that appeared to be so simple that even I couldn't resist changing a single resistor, pardon the pun. According to the article, TI implemented a 56Ø ohm resistor as the VDP Load Resistor, while TMS-9918 specs called for a 33Ø ohm resistor. Lowering the resistor value yields faster fall times, which results in a sharper picture. The effect is to reduce white shadows.

Before getting started, you'll need some items that are rather standard for hardware hacking. A small Phillips screwdriver, a soldering iron, a desoldering device, a pair of diagonal cutters and some long-nosed pliers will make this project easy. You'll also need a little solder. If you are unfamiliar with the use of the tools, you should find someone who is experienced to help you or do the job for you. While this is a very simple project for anyone familiar with soldering, wielding a hot iron inside the console can spell trouble for the uninitiated. If you are a beginner and you can't find anyone to help you, then I recommend that you do your practicing on a spare (expendable) console if you have one. Even the simplest of hardware projects can lead to a malfunctioning computer and this one is no exception.

The resistor that you'll need is 33Ø ohms (that's not 33Øk ohms!), 1/4 watt (resistors are usually 1/2 or 1/4 watt). At Radio Shack, the part number is 271-1315 and a pack of five is 39 cents! For pennies you could while away your time making this mod for everyone in a small to medium-sized user group!

When disassembling the console, take note of how everything fits together as it is necessary to put it back together to get it to work properly! Ha! Open the console, being very careful so as not to damage the on/off switch. Inside you'll find the keyboard, a small power board near the on/off switch and an RF shield metal case. Yes, the main board on which the VDP is located is inside the metal case, more unscrewing. Before taking the case apart, remove the GROM connector from its port (yes, the connector right angles into another port).

Locate the VDP chip on the main board. It is one of the larger chips on the board. Unfortunately, any numbers that might otherwise be recognizable on the VDP may not be readable, as the VDP is often coated with heat sink compound. Actually, a bank of 4116's, the VDP memory, is the best landmark for finding the general area where the resistor can be found. Refer to the simple diagram below or a good set of console schematics if you have any doubts.

Trace Pin 36 on the TMS 9918 VDP through two inductors to the VDP Load Resistor to ground. If you have a Technical Reference Manual, the resistor is identified as R212. Be sure to determine that the resistor is 56Ø ohms (resistors are color-coded, refer to a chart if you aren't familiar with the color system).

After locating the resistor to change, turn the board over and desolder the resistor. Pull it out, put the new one in, clip the resistor leads to just long enough to be accessible for soldering, bend the ends of the leads in toward the board and solder in place. Simple!

Yes, I've probably gone into far more detail than many of you required, but I hope that this article will appeal to those who haven't yet caught "hardware fever". While this mod does seem to do some good, I can't really say that it is particularly spectacular in terms of effort versus results. If you've ever taken a console apart, you probably know what I mean. This project will probably take 2Ø minutes to 2 hours, depending on your experience level. But, it does give the beginner an opportunity to undertake a project and, besides, while you're inside the console you can do a little cleaning!

Here's a crude diagram (shapes should not be considered an indicator of component type, but rather just the approximate size of the components) showing how the area around the VDP looks in many consoles (the exact locations of components may vary):

```
            |--|

            |--|        +-----+  +-----+
                        |     |  |  T  |
                        +-----+  |  M  |
|----|  |----|                   |  S  |
                        +-----+  |     |
|----|                  |     |  |  9  |
        |\   |\         +-----+  |  9  |
|----|  |/   |/                  |  1  |
                      |--|  |-|  |  8  |
|----|                           |     |
                      |--|  |-|  |  V  |
|----|  |----|                   |  D  |
                      |--|+---+  |  P  |
|----|  |----|            |   |  |     |
         R212             +---+  |     |
         load                    |     |
       resistor                  |     |
                                 |     |
                                 |     |
    +--------+          +----+   |     |
    |        |          |    |   |     |
    |        |          |    |   |     |
    +--------+          +----+   |     |
                                 +-----+
                         |----|

 +-+ +-+ +-+ +-+ +-+ +-+ +-+ +-+
 |4| |4| |4| |4| |4| |4| |4| |4|
 |1| |1| |1| |1| |1| |1| |1| |1|
 |1| |1| |1| |1| |1| |1| |1| |1|
 |6| |6| |6| |6| |6| |6| |6| |6|
 | | | | | | | | | | | | | | | |
 | | | | | | | | | | | | | | | |
 | | | | | | | | | | | | | | | |
 +-+ +-+ +-+ +-+ +-+ +-+ +-+ +-+
```

(the 4116's are video memory)

FCC Proposals Have Far-Reaching Effect

by Richard M. Mitchell

The Federal Communications Commission has drafted a proposed regulation, scheduled to take effect January 1, 1988, that would redistribute the source of revenues of the phone companies and have the effect of hiking the costs of telecommunications networks substantially. It is anticipated that the cost increases, expected to be approximately $5.ØØ per hour, will be passed on to consumers. For 4A users, specifically affected would be networks such as The Source, GEnie, Compuserve and Delphi and similar services would also be affected under the regulation. If the proposal is implemented, it would likely drastically reduce accesses by individuals and effectively deteriorate a prime medium for freedom of speech.

If you are concerned, as I am, that such drastic measures will have a profound negative impact on the educational, economic and intellectual future of the USA, I urge you to send your comments to:

FCC
Secretary's Office
1919 M Street N.W.
Room 222
Washington, D.C. 2Ø554

You may also write to the Chairman and Commissioners at the above address:

Honorable Dennis R. Patrick
Chairman, FCC

Honorable James H. Quello
Commissioner, FCC

Honorable Patricia Diaz Dennis
Commissioner, FCC

Honorable Mimi Danforth Dawson
Commissioner, FCC

In your correspondence, refer to "Interstate Access Charges Exemption for Enhanced Service Providers FCC Docket 87-2Ø8". A copy of the order can be obtained by calling the FCC at (2Ø2) 632-7ØØØ (the FCC, by law, must respond to citizen inquiries).

Accessing Two Registers With a Single Operand

by Richard M. Mitchell

While many of you may be familiar with accessing two registers by using instructions such as MOV that employ two operands, you may not be aware that single operand instructions can also access two registers!

The key to the trick is to store the address of a register in a register. That can be done with an EQUate. Here's an example of incrementing a register as a counter and acting on the contents of the address in the register, which is the address of another register:

```
CFI     EQU  >12B8
ERRBV   EQU  >1EØØ
NUMREF  EQU  >2ØØC
XMLLNK  EQU  >2Ø18
ERR     EQU  >2Ø34
FAC     EQU  >834A
GPLWS   EQU  >83EØ

START
        LWPI MYWS
        CLR  RØ        NO ARRAYS
        LI   R1,1      START AT PARM 1
        LI   R3,REG4   START AT R4
        LI   R1Ø,REG9  STOP BEFORE R9
START1  BLWP @NUMREF   PARM AT FAC
        BLWP @XMLLNK   USE ROM CODE
        DATA CFI       FLOAT...INTEGER
        MOV  @FAC,*R3  FAC TO R4...R8
        CI   *R3,Ø     EQUAL TO Ø?
        JEQ  STRERR    YES, BAD VALUE
        CI   *R3,>2Ø   NOT OVER 32?
        JLE  START2    OK, JUMP
STRERR  LI   RØ,ERRBV  BAD VALUE
        BLWP @ERR      SEND ERR TO XB
START2  DEC  *R3+      DEC, NEXT REG!
        INC  R1        NEXT PARM
        C    R3,R1Ø    LAST PARM?
        JNE  START1    NO, JUMP
        LWPI GPLWS     READY FOR XB
        B    @>ØØ6A    RETURN TO XB

MYWS    BSS  >2Ø
REG4    EQU  MYWS+8
REG9    EQU  MYWS+>12
```

The above routine will get five numeric parameters from an Extended BASIC LINK and place those values in Registers 4 through 8. While the specific values being checked implies that the numbers may be screen rows, subsequent code for utilizing the values is beyond the scope of this article. The line of interest for this article is the code at the label START2. Because R3 was loaded with the address of a register, the program can access consecutive registers in a loop and simultaneously act on the contents of those registers! As the loop progresses, "DEC *R3+" DECrements by one the contents of R4 and then sets R3 to point to R5, on the next loop cycle DECs R5 and then points to R6, and that continues until the fifth value is handled at R8! That's a lot of activity for a relatively small Assembly routine!

To obtain other results, the DEC in the example program could be replaced with other word-oriented instructions, such as DECT, INC, INCT, SWPB, etc., for some powerful manipulations.

Oops!

In the April, 1984 issue of *The Smart Programmer*, at the bottom of page 1Ø, "EXAMPLES with OPTION BASE Ø", the elements listed in the string examples are inaccurate and should be as follows:

```
DIM S$(2,2) {type >82}

Byte
No.
 7- 8 = >Ø2
 9-1Ø = >Ø2
11-12 = VDP address of S$(Ø,Ø)
13-14 = VDP address of S$(Ø,1)
15-16 = VDP address of S$(Ø,2)
17-18 = VDP address of S$(1,Ø)
19-2Ø = VDP address of S$(1,1)
21-22 = VDP address of S$(1,2)
23-24 = VDP address of S$(2,Ø)
25-26 = VDP address of S$(2,1)
27-28 = VDP address of S$(2,2)
29-3Ø = >5324 = S$
```

The element numbers of multi-dimensional numeric arrays are also sequenced as shown above, but utilize the eight-byte High Memory structure described in the 1984 article. Yes, the E/A manual's description of the numbering sequence is ambiguous.

SWAP BYTE Bites

by Richard M. Mitchell

One of the major objectives for most 99/4A Assembly programmers is the wise usage of memory, as there is not an abundance available and sloppy Assembly code can devour many bytes needlessly. Luckily, the 99/4A provides the opportunity to easily access routines repeatedly and to use a few other tricks to conserve bytes, thus allowing for powerful programs to be written in a very small block of memory.

Often, programmers fail to recognize when codings can be performed directly rather than routing values redundantly through registers. Also, the specific instructions used can make a difference. While "only" 4 or 6 or 8 bytes may not seem like much in the context of a single routine, saving 4 bytes per 8 lines of source code across an entire program can leave adequate space for avoiding "slacking"; include those error traps!

One example of commonly used inefficient coding is access to the Least Significant Byte (LSB) of a Word of memory. It is not mandatory to move the byte to a register to access the LSB and doing so unneccessarily can put a stranglehold on usage of the workspace. Additionally, it is not always necessary to use SWPB, as a byte can be addressed with a label! Yes, SWPB is sometimes required, but two SWPBs on a Word without an intervening Word operation is generally a sure sign of inefficient coding. Here are four examples that illustrate the hazards mentioned and some alternate methods:

```
* METHOD 1
* R4 not used, SWPB is inefficient
* TOTAL OF 54 BYTES

START
        LWPI MYWS
        SWPB @MAX
        MOVB @MAX,@LENBYT
        SWPB @MAX
        RT

MYWS    BSS  >2Ø
MAX     DATA >1ØFF
LENBYT  BYTE Ø
        EVEN

* METHOD 2
* Efficient if >FF1Ø is to be used
*    subsequently from R4
* TOTAL OF 5Ø BYTES

START
        LWPI MYWS
        MOV  @MAX,R4
        SWPB R4
        MOVB R4,@LENBYT
        RT

MYWS    BSS  >2Ø
MAX     DATA >1ØFF
LENBYT  BYTE Ø
        EVEN
```

```
* METHOD 3
* R4 not used.
* SWPB not used.
* Label access to LSB for clarity.
* TOTAL OF 46 BYTES

START
       LWPI MYWS
       MOVB @MAX,@LENBYT
       RT

MYWS   BSS  >20
SIXTEN BYTE >10
MAX    BYTE >FF
LENBYT BYTE 0
       EVEN
```

```
* METHOD 4
* LSB already in R4
* Label access to LSB of register
* SWPB not used
* TOTAL OF 44 BYTES

START
       LWPI MYWS
       MOVB @LSB4,@LENBYT
       RT

MYWS   BSS  >20
LSB4   EQU  MYWS+9
LENBYT BYTE 0
       EVEN
```

Many more examples could be provided. Clearly, the most efficient method is dependent upon the code that will subsequently surround the program shell shown. If usage of registers and SWPB can be avoided, then several bytes can often be saved.

As there are dozens of ways to code access to the LSB of a Word, there are certainly many optional methods of writing an entire program! Never get discouraged! Even the best programmers sometimes write lousy sections of code. The important factor is that a program, taken as a whole, should be efficient, fast, error-trapped and user-friendly (yes, those objectives do often conflict, so a balanced trade-off is frequently required). The perfect program will never be written, but with practice a programmer can write code that represents a personal achievement, a good goal for us all!

BASIC Tips

by Stephen Shaw
Extracted from The Book by permission of the author, Stephen Shaw
Copyright 1983 Stephen Shaw. All Rights Reserved.

A=B^2 takes longer to process than **A=B*B.**

A=20000 uses more program memory than A=2E4.

Editor: More extracts from Stephen Shaw's book will be featured in future issues.

Letter & Comments

by Richard M. Mitchell

I recently received a lengthy note from Stephen Shaw that included some important observations. Here's a portion of what Stephen had to say:

> *It fills me with considerable distress to see fellow TI owners spending their time with destructive criticism of other TI owners and supporters -- when all the producers have been chased away, no one will replace them.*
>
> *The problem is a simple one of communication: those who are supporting our family are failing to tell us under what conditions they are operating.*

The comments are applicable to all of our support arms -- user groups, vendors, telecommunications sysops, programmers, etc. Nearly all of the folks operating in support capacities are either unpaid or are making far less profit than they are qualified to earn. Almost without exception, they're doing it because they enjoy using the 4A. And, considering that the 4A community's successes today are due almost exclusively to the efforts of volunteers and "cottage businesses" ("cottage businesses" operate from private homes and I herein include fairware vendors in this category), things are really going great!

So, I hope we'll all make an effort to get things rolling on a positive note. A little encouragement goes a long way! Merely evaluating inadequacies is not enough;

solutions count! So, do provide constructive criticism! Most folks will appreciate the help!

Many times I'm asked why commentary in this newsletter consists primarily of positive remarks. Well, I simply choose not to provide notoriety to the mediocre when I can use the same space to laud the outstanding. Think about it.

5TH 1- =FORTH

by Mariusz Stanczak

Well, it was bound to happen sometime. What a mess, and whatever the reason, the last article was infested with so many bugs and unprintable characters, that it will take us a large portion of today's meeting to clear them out, and to bring back some sense to what will remain.

First of all, if, after reading the last column, you had wondered why the PSEUDO83 listing started with the line 99, and what had happend to the lines 1-98; well, they were contained in a lost column that was supposed to precede the column about MODULE. Those missing lines you will find below, with the original introduction that accompanied them. Then we'll take care of bugs in MODULE, and, lastly, we'll have an example of how the classic game of Towers of Hanoi could be written using recursive technique, so let's get to it.

Editor: From the beginning, I promised Mariusz complete editorial freedom in this column, as I had little personal interest in FORTH. While endeavoring to abide by that policy, I overlooked rudimentary editorial responsibilities, such as checking that all articles were received through the mail. Yipes! My apologies to everyone, especially Mariusz. Richard M. Mitchell

Mr. Gene Thomas, a member of a very active Central Arkansas FIG Chapter, had mailed to us a listing of words that make it possible to run, with little or no modifications, Forth-83 programs on FIG systems. I'm not a great fan of that standard but the code is very nicely written and you may find it attractive to have a vocabulary that will give you the benefit of typing in directly the code that's published in magazines and recent books. There are about twenty screens equally divided among the conversion words and extensions to Forth-83 standard built upon those words, so we'll publish them in three installments.

```
 0.S; PSUEDO83 -- Conversion of FIG to Forth-83, with extensions
 1.        ver 2.1 corrects some bugs and expands ver 2.0
 2.    Contains some public domain material from other sources
 3.              (c)  copyright 1986 by Gene Thomas
 4.          Permission is granted to the public domain
 5.          for private and commercial use provided
 6.          that the following statement accompanies
 7.          its use, and if only a few sporadic words
 8.          are used, the ID, gtPsuedo83, be placed in
 9.          a comment with each word used.   Public
10.          domain material of others is indicated in
11.          the listing.  The statement for inclusion is:
12.
13.          This application contains some or all of
14.             PSUEDO83: COPYRIGHT 1986 by Gene Thomas
15.                   Used with permission.
```

```
\ Load scr: Psuedo83, synonym see FD V.8/1       ver 2.1 gt Jul86
: SYNONYM ( VHY 11/28/84; convert to fig gt Apr86)
   <BUILDS  -FIND
   IF  DROP DUP CFA , IMMEDIATE NFA 64 AND
     IF  DOES>  @ EXECUTE
     ELSE  DOES>  STATE @
       IF  @ ,
       ELSE  @ EXECUTE
       THEN
     THEN
   ELSE  CR ." Not found" ABORT
   THEN
; \ a public domain word
: MORE-LOADS ( n --|load n more screens from here)
   BLK @ 1+ DUP ROT + SWAP  DO  I LOAD  LOOP
;

\ Psuedo83  '83, ['], >body, word,               ver 2.1 gt Jul86
: PSUEDO83 ; \ a noop marker
: '83  ( -- cfa|equivalent to forth 83's ' {tick})
   -FIND 0=  IF  ." Not found" QUIT  THEN  DROP 2-
;
: [']  [COMPILE] '83  [COMPILE] LITERAL
; IMMEDIATE
\ The reminder of this block is not copyrighted.
\ Following are based on the dictionary order: lfa nfa cfa pfa
: N>LINK  ( nfa -- lfa)
   PFA LFA ;
: LINK>  ( lfa -- cfa)
   2+ PFA 2- ;
: >LINK  ( cfa -- lfa)
   2+ LFA ;
: BODY>  ( pfa -- cfa)
   CFA ;
: >NAME  ( cfa -- nfa)
   NFA ;
: L>NAME  ( lfa -- nfa)
   2+ ;
: NAME>  ( nfa -- cfa)
   PFA 2- ;
: >BODY  ( cfa -- pfa)
   2+ ;
: VARIABLE83  \ no stack argument
   <BUILDS  0 ,  DOES> ;
: WORD83  WORD HERE ;

\ Psuedo83  .(, find                             ver 2.1 gt Jul86
\ FIG's -find and 83's find are not much alike
: FIND  ( adr -- cfa f|found f=1 if immediate, otherwise -1)
        ( adr -- adr 0|not found; adr unchanged)
   DUP ( save adr)  CURRENT @ @  (FIND)
   IF ( found) ROT DROP DROP DUP ( drop adr & count, dup pfa)
     2- ( pfa->cfa)  SWAP NFA C@ 64 AND ( immediate?)
     IF  1  ELSE  -1  THEN
   ELSE  0 ( not found)
   THEN
;
\ remainder of block not copyrighted
: ASCII ( -- n|takes following char,_leaves ascii code on stack)
   BL WORD83 1+ C@  [COMPILE] LITERAL
; IMMEDIATE
```

''NOTE The above definition is useful while compiling but it will not work properly from the keyboard. Forth-83 specifies words that do work in either mode as 'state smart' and after Thinking Forth (L. Brody '84) ASCII word that exhibits such behavior could be defined as:

```
: ASCII ( -- c)
        \ Compile: c ( -- )
        \ Interpret: c ( -- c )
    BL WORD83 1+ C@ STATE @
    IF [COMPILE] LITERAL THEN
; IMMEDIATE
```

Some of you may prefer the latter version, as it permits its use interactively from keyboard. ''END-OF-NOTE

```
 75.: .( ASCII ) WORD83 DUP C@ CR TYPE
 76.; IMMEDIATE
 77.: PICK83 1- PICK ;
 78.: ROLL83 1- ROLL ;
 79.: CONVERT (NUMBER) ;
 8Ø.: NEGATE MINUS ;
 81.: DNEGATE DMINUS ;
 82.: ?DNEGATE D+- ;
 83.SYNONYM ?NEGATE +-
 84.SYNONYM UM* U*
 85.SYNONYM .NAME ID.
 86.SYNONYM CREATE83 <BUILDS
 87.SYNONYM >IN IN
 88.SYNONYM RECURSE MYSELF
 89.SYNONYM S>D S->D
 9Ø.SYNONYM LAST83 LATEST
 91.SYNONYM ?DUP -DUP
 92.SYNONYM ?BRANCH ØBRANCH
 93.SYNONYM EXIT ;S
 94.SYNONYM UM/MOD M/MOD \ unsigned #'s, flooring unnecessary
 95.: R@ R> R SWAP >R ;
 96.\ some words (like DNEGATE) will not operate within control
 97.\ structures (if-then) if defined with synonym.  Others?
 98.\ This problem may be unique to this implementation
```

So far, we've covered mostly redefinitions with the exception of FIND and SYNONYM. It is quite a feat to simulate Forth-83 FIND in Fig-Forth; so naturally, the high level implementation is awfully inefficient. For those swift in assembly I'd recommend recoding it. SYNONYM is great. It has no overhead during execution and can make your code read like English (well, almost). If you have access to *Forth Dimensions* (the subscription rate went up again to $3Ø.ØØ/year but, if you can spend this much, subscribe!) read Mr. V.H. Yngue's two part tutorial Synonyms and Macros in V.7/3 and Benchmark Readability in V.7/4

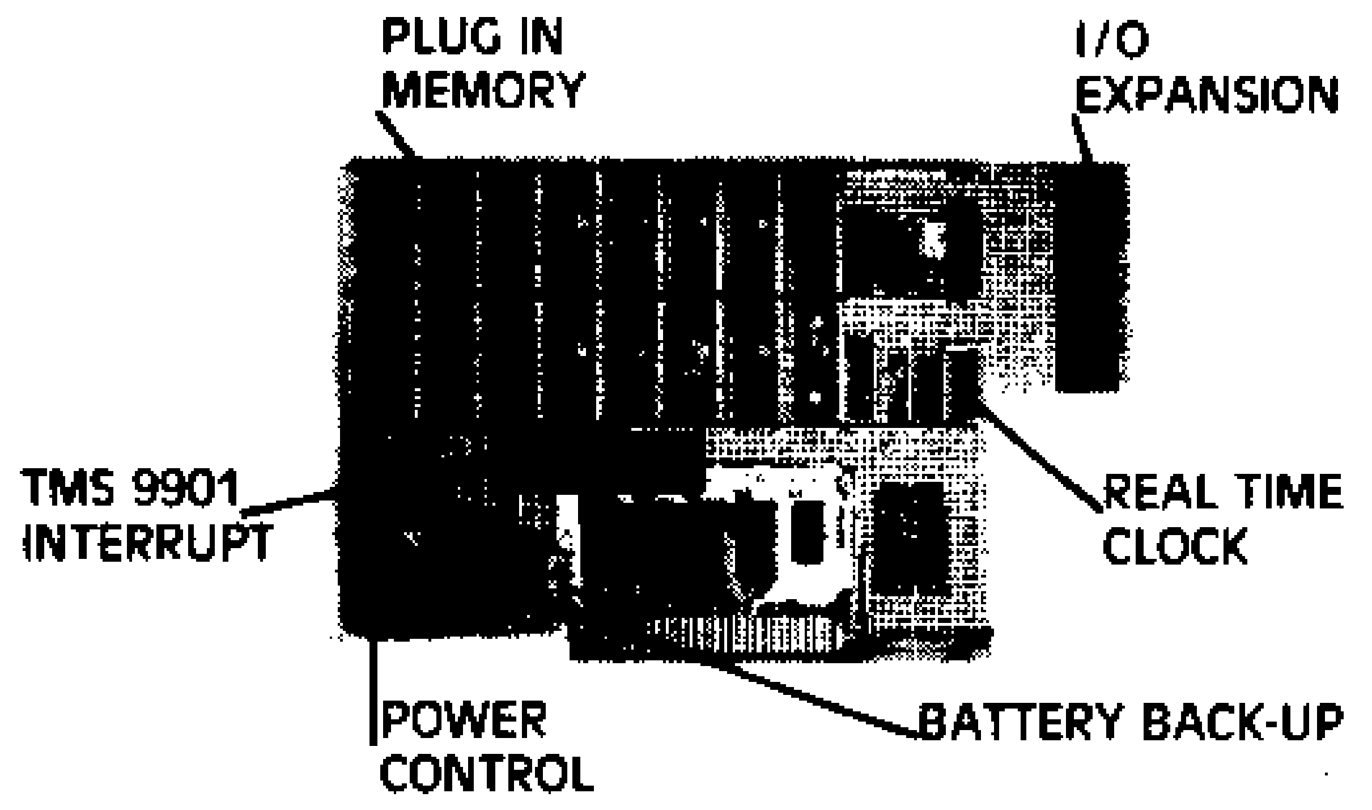
In the MODULE article, the errors were many (who am I telling this!), so I'll give you the line number and its correct form. The correction lines below replace the lines as they originally appeared, and the numbers refer to the number of the code line as it appeared.

The $129.95 Bombshell

GRAND RAM SYSTEM EXPANSION

A NEW AND EXCITING OPTION FOR YOU AND YOUR TI 99/4A

GRAND RAM



GRAND RAM FEATURES

- Print spooler keeps system active while printing.
- Battery backed to eliminate data loss and power supply.
- Optional real time clock.
- Plug-in memory for easy expansion up to 512K.
- Compatible with TI, Myarc, Geneve, CorComp, Morning Star, Foundation.
- I/O expansion for future products.
- TMS 9901 on-board interrupt to speed input/output.
- Fully documented source code.
- Accessible from any language.
- Selectable disk drive configuration gives you up to 4 drives per Grand Ram.
- Disk manager and terminal emulator software.

SUPER ELECTRONICS

Los Angeles — DataBiotics has announced the release of what it terms "super electronics" for the TI 99/4A computer. DataBiotics spokesmen describe the device as "the ultimate plug-in" enhancement, not surprisingly named **GRAND RAM. GRAND RAM,** a drop-in card for the expansion box, is described as a super RAM disk with nearly limitless growth potential.

WHAT'S IN A NAME?

Does **GRAND RAM** live up to its name? Picture a disk drive that loads and saves 6 times more data 10 times faster than your drive, has no moving parts to fail, and probably costs less than your old drive. Then, picture being able to turn this double-sided, double-density drive into 4 single-sided or 2 double-sided drives in just minutes to make multiple copies simultaneously and do file management. Picture being able to simply plug in memory chips when you need to expand, and when you can afford to expand. Picture being able to use this device with BASIC, EX-BASIC, PASCAL, "C" or any other language, knowing you won't lose your work ever because the device is battery-backed!

```
****************************
* SHIPPING SEPTEMBER 1987 *
****************************
```

THE AFFORDABLE COLOSSUS

The real appeal of **GRAND RAM** to many users may be extraordinary expandibility. User groups have been standing in line for **GRAND RAM** for nearly two years, and many have prepaid for early delivery. The truth is that **GRAND RAM** is an affordable colossus. Buy a basic version with a small cash outlay and periodically plug in more memory or accessories as you need them!

GROWTH UNLIMITED

Are there accessories? How about a plug-in real time clock? And available soon, a device called analog-to-digital to run external equipment and an emulator to create your own cartridges?* Need more memory? Just plug in chips to expand up to 512K of memory. That's 200 pages of text that your device can hold! Or buy more **GRAND RAM** units to expand up to 2 megabytes! And **GRAND RAM** is compatible with CorComp, TI, Myarc, Geneve, Morning Star, Foundation and with the John Johnson menu!

SO WHAT DO I NEED TO START?

You need an expansion box and a disk controller. We give you a superb disk manager and 4A/TALK terminal emulator to communicate with the world. You also get fully documented **GRAND RAM** source code for the device, print spooler and loader/configurator, and complete instructions.

*Available March 1988

64K	ORDER NO. DB646R	$129.95
128K	ORDER NO. DB128R	$144.95
256K	ORDER NO. DB256R	$169.95
512K	ORDER NO. DB512R	$229.95
CLOCK-	ORDER NO. DBCLGR	$ 19.95

California residents add sales tax.
($3.00 postage & handling).

TECHNICAL HOTLINE
In California: 707-585-3922
Outside California: 1-800-255-2985

GRAND RAM KIT (Build Your Own)

64K	$ 99.95
128K	112.95
256K	136.95
512K	185.98
CLOCK CHIP	$19.95

(If ordered with kit or built-up model)

```
Listing, Part 1:
19. DP>MODULE_BUF ;
24. DUP 4 TH ADR @ R - R> DUP B/BUF + >R
36. [COMPILE] : \ in Wycove's [COMPILE] (:)
37. COMPILE OFFSET COMPILE DUP COMPILE @
38. COMPILE LIT HERE 9 TH POINTER! 0 ,
39. COMPILE 3 COMPILE PICK COMPILE !
40. COMPILE LIT HERE 8 TH POINTER! 0 ,
41. COMPILE MODULE_LOAD COMPILE SWAP COMPILE !
42. [COMPILE] ; \ in Wycove's [COMPILE] (;)
45. HERE 4 TH POINTER! VOC-LINK @ 6 TH POINTER! DP !
46. MODULE_LINK: LATEST PFA CFA DUP @ 3 RD POINTER! 2 ND POINTER! ;
47. : MODULE ( any_#_of_screen_#s ---) \ followed by inline name
54. 5 TH POINTER@ 6 TH POINTER@ OVER OVER -
60. DROP SWAP DROP DUP 7 TH POINTER! !
65. DUP 8 TH POINTER@ ! OFFSET @ 9 TH POINTER@ !
66. DICTIONARE_RELINK VOCABULARY_RELINK
69. >R DUP 1+ SWAP OFFSET @ + BUFFER UPDATE
71. 4 TH POINTER@ R - R> DUP B/BUF + >R
72. SWAP >R SWAP R> B/BUF MIN CMOVE
77. Environment info parameter array. <> means the content of
78. 1 ST - old_latast's nfa (in FORTH vocabulary)
85. 8 TH - adr in parameter field of module_name
86. 9 TH - adr in parameter field of module_name
```

```
Listing, Part2, Pseudo83
178. IF I @ 8000 - I 2+ SWAP 0 R/W I !
```

There is one correction to the text of the article and the error was mine. In the decription of CLF (top portion of page six) I should have said, "points to CLF of previous (and not parent) vocabulary. The reason why this field is called "chronological" is because this field links all vocabularies in the system according to the order in which they were created, and not to which vocabulary they belong. The latter function, as we all know by now, is served by the user variable VOC-LINK.

OK! Now that we have, hopefully, cleaned up the old errors, let's jump right into making some new ones (just kidding!), because it is just about time we start something new. In the past few months, I was drawn aside into other projects, but the last thing I played with on the 99/4A was a very neat expert system shell called FORPS. We'll run the code (three screens in all, and it's a real screamer) in the next column, but today, for comparison with the example that the authors of the original paper on FORPS published, we'll look at an implementation of the game Towers of Hanoi as it was done by S.D. Roberts (plus my quite extensive modifications) in his book FORTH Applications (Bitfire series, 1985, Elcomp Publishing, Inc.). The code may seem rather cryptic, but only if you don't know the algorithm, so read on. Given three pegs, A, B, and C, and N disks stacked up on peg A, each one smaller than the one below it, the object of this ancient game is to move all disks from peg A to peg B using peg C, one disk at a time, and in a such a way that a larger disk is never put on top of a smaller one. Simple! Those are the rules of the game, and the principle for the algorithm is as follows:

1) Move N-1 disks from peg A to peg C.
2) Move bottom disk from peg A to peg B.
3) Move N-1 disks from peg C to peg A.
4) Move bottom disk from peg C to peg B
5) Repeat steps 1 to 4 until all disks are moved to peg B

The procedure to move N disks is to move N-1 disks twice, which is accomplished by moving N-2 disks twice, and so on. After each cycle we end up with N-1 disks on either peg A or on peg C, which then become our next level's N, until all disks are moved. Let's call our top level ABCN (which means move, from A to B using C, N disks); it will be 'called' once, and since N<>Ø, ABCN will invoke ACBN-1 until there is only one disk left on peg A (actually the test is "Ø> IF," but since N-1 disks were moved, there is one left), and that one will be moved by A>B (actually it will be either A to B or C to B). After that is accomplished, N-1 (our new N) disks are on peg C and the recursive cycle starts all over.

```
: 2DROP   DROP DROP ;
: 4DROP   2DROP 2DROP ;
: 4DUP   4 Ø DO  SP@ 6 + @  LOOP ;
```

** NOTE - The 4DUP could be made into a more general n NDUP tool by not hardcoding the upper loop limit nor the offset into the stack, as if

```
: NDUP ( n -- n1 n2 ... n)
   DUP HERE !  \ temporary storage for the number of items to be duplicated
   Ø DO  HERE @ 2 * SP@ + @  LOOP ;
```

which, for debugging, could be supported with ?ENOUGH. END-OF-NOTE **

```
: .PEG
    CASE
       1 OF ." A" ENDOF
       2 OF ." B" ENDOF
       3 OF ." C" ENDOF
    ENDCASE ;
: CBAN-1 ( n1 n2 n3 n4 -- n3 n2 n1 'n4)
    4DUP 1-  >R SWAP ROT R> ;
: A>B 4DUP 2DROP SWAP CR ." Move disk on peg " .PEG
    ."  to peg " .PEG ;
: ACBN-1 ( n1 n2 n3 n4 -- n1 n3 n2 'n4)
    4DUP 1- >R SWAP R> ;
: ABCN ( n1 n2 n3 n4 --)
    DUP Ø>
    IF  ACBN-1 MYSELF  A>B CBAN-1 MYSELF
    THEN  4DROP ;
: TOWER ( n --)
    DEPTH Ø=
    IF  CR ." Re-execute with the number of disks on stack" QUIT  ENDIF
    >R 1 2 3 R>  ABCN  CR CR ." Finished" ;
```

I encourage you to carefully trace the word ABCN with, lets say, 4 disks; after which, you should have a very firm grip on the use of the word MYSELF, the Forth's way to permit coding of recursive routines. A way maybe not as straightforward as in, lets say, Pascal or C, but nevertheless as flexible, if not more (after you get the hang of it). Have fun, and may Forth be with you!

Traffic Light

by Richard M. Mitchell

From time to time, I enjoy including a program for children, so here's one that teaches the sequence of green, yellow, red at a traffic light. My four-year-old really enjoys it, so I hope others will, too. You will need expanded memory (or Triton Super Extended BASIC, which allows CALL LOAD's without expansion memory) and a color monitor to make use of this one. Note that the lights must be accessed in the proper sequence. As promised last issue, checksums are included.

```
> 100 CALL INIT :: CALL MAGNIF
  Y(2):: Z=14 !199
> 110 CALL CHAR(143,"0000183C3
  C18",142,"FFFFE7C3C3E7FFFF",
  141,"000003043F7FFF")!179
> 120 CALL CHAR(140,"000000010
  020502",139,"0000C020FCFEFF"
  ,138,RPT$("0",11)&"40A04",13
  7,RPT$("80",8),136,"FF")!045
> 130 DISPLAY AT(20,1)ERASE AL
  L:"PRESS 1) FOR RED":"
  2) FOR YELLOW":"       3) FOR
   GREEN":"       4) TO CHANGE
  CAR COLOR":"       5) TO END"
   !109
> 140 CALL HCHAR(16,1,136,4)::
   CALL HCHAR(16,9,136,24):: C
  ALL HCHAR(19,1,136,32):: CAL
  L VCHAR(1,9,137,15):: CALL V
  CHAR(1,5,137,15)!009
> 150 CALL SPRITE(#4,142,13,76
  ,43,#5,143,2,76,43,#6,142,13
  ,92,43)!030
> 160 CALL SPRITE(#7,143,2,92,
  43,#8,142,13,108,43,#9,143,2
  ,108,43)!079
> 170 CALL LOAD(-31806,64):: C
  ALL SPRITE(#11,141,14,120,56
  ,0,-25,#10,140,2,120,56,0,-2
  5)!240
> 180 CALL SPRITE(#13,139,14,1
  20,72,0,-25,#12,138,2,120,72
  ,0,-25):: CALL LOAD(-31806,0
  )!195
> 190 CALL SPRITE(#1,143,2,76,
  43,0,0,#2,143,2,92,43,0,0,#3
  ,143,4,108,43,0,0):: K1=3 !1
  60
> 200 CALL KEY(5,K,S):: CALL P
  OSITION(#10,Y,X):: IF K1>2 T
  HEN CALL LOAD(-31806,0)ELSE
  IF (X<100)AND(X>61)THEN CALL
   LOAD(-31806,64)!100
> 210 IF K1=3 THEN CALL SOUND(
  -700,110,30,110,30,110,30,-8
  ,0)!224
> 220 IF (S<1)OR(K<49)OR(K>53)
  THEN 200 !010
> 230 C=K1 :: K1=K-48 :: ON K1
   GOSUB 240,260,280,300,290 :
  : IF K1<5 THEN 200 ELSE END
  !063
> 240 IF C=2 THEN CALL COLOR(#
  1,7,#2,2,#3,2)ELSE CALL SOUN
  D(100,110,0):: K1=C !219
> 250 RETURN !136
> 260 IF C=3 THEN CALL COLOR(#
  1,2,#2,11,#3,2)ELSE CALL SOU
  ND(100,110,0):: K1=C !008
> 270 RETURN !136
> 280 IF C=1 THEN CALL COLOR(#
  1,2,#2,2,#3,4)ELSE CALL SOUN
  D(100,110,0):: K1=C !215
> 290 RETURN !136
> 300 Z=Z+1 :: IF Z>16 THEN Z=
  3 ELSE IF Z=8 THEN Z=Z+1 !06
  1
> 310 CALL COLOR(#11,Z,#13,Z):
  : K1=C :: RETURN !128
```

Q & A

What is a pointer?

The term "pointer" is often used rather loosely to refer to any location that points to another location. The possibilites for locations pointed to include a memory address, a file record, an item of data, a screen location, etc. Typically, the pointer is at a fixed location, while the location referenced can vary. TI used many memory pointers in the 4A to provide programming ease and to ensure compatibility among operating system releases (yes, there were at least seven console variations). It is important that distributed software be usable on any 4A, so the pointers are very important. Directly addressing ROM and GROM may not locate the proper address in all consoles, so pointers should be used. Where a pointer to an address cannot be located, the next safest approach is a search for the code that is to be accessed. Users with hardware that allows changing ROM and GROM code should be aware that the 4A has a complex system of pointers and indiscriminate changes can wipe out the compatibility that users rely upon.

Do you recommend using user-written subprograms in XB?

Subprograms are quite useful for structuring and can be used to save bytes. However, the manner in which XB accesses subprograms is extremely slow. Subprograms can utilize variables in such a way as to execute more quickly than a subroutine, but such cases are rare. If a program requires complex logic and execution speed is insignificant, subprograms may be the best route. But, for applications where speed is critical, consider subprograms only as a last alternative. To test execution speeds, time a FOR-NEXT loop, then time executions within the loop and subtract the time it takes to execute the loop only.

BYTEMASTER ORDER FORM

NAME________________

ADDRESS________________

CITY________________

STATE________________

ZIP CODE________________

COUNTRY________________

The Smart Programmer

SP1	$18.00	U.S. AND CANADA FIRST CLASS
SP2	$15.00	U.S. THIRD CLASS (no back issues)
SP3	$20.00	FOREIGN SURFACE (no back issues)
SP4	$32.00	FOREIGN AIRMAIL
SP5A-G	$ 1.75	U.S. JUNE - DEC 1986, ea.
SP6A-G	$ 2.75	FOREIGN JUNE - DEC 1986, ea.
SP7A-G	$ 2.50	U.S. JAN 84-AUG 84,photostats,ea.
SP8A-G	$ 3.50	FOREIGN JAN 84 - AUG 84, ea.

Super 99 Monthly

SM1	$18.00	Complete set of 18 back issues
SM2A-R	$ 1.00	Back issues - ea. (U.S. Third Class)
SM3A-R	$ 1.50	Back issues - ea. (Canada and U.S. First Class)
SM6A-R	$ 2.50	Back issues - ea. (Foreign Air Mail)
SM4	$12.00	Programs on disk (non-FORTH)
SM5	$15.00	Super 99 Handicapper (thoroughbreds)
SM6	$24.95	Explorer
SM7	$19.95	String Master

ITEM #	QTY	EACH	AMOUNT	New	Renewal
______	______	______	______	□	□
______	______	______	______		

Payments accepted by check or money order in U.S. funds, coded for processing through the U.S. Federal Reserve Banking System. No billings or credit sales. Dealer inquiries invited. Discounts available on quantity orders.

Software shipping and handling: $2.00 US and Canada, $6.00 other. Louisiana residents must add 4% sales tax. Calcasieu 1%. Sulphur 2%.

The Smart Programmer is published monthly by Bytemaster Computer Services, 171 Mustang Street, Sulphur, LA 70663. All correspondence received will be considered unconditionally assigned for publication and copyright and subject to editing and comments by the Editor of *The Smart Programmer*. Each contribution to this issue and the issue as a whole COPYRIGHT 1986 by Bytemaster Computer Services. All rights reserved. Copying done for other than personal archival or internal reference use without the permission of Bytemaster Computer Services is prohibited. Bytemaster Computer Services assumes no liability for errors in articles.

Editor Richard M. Mitchell
Staff Craig Miller
Charles M. Robertson
Mariusz Stanczak
Steven J. Szymkiewicz, MD
Barry A. Traver
D.C. Warren

Bytemaster Computer Services
171 Mustang Street
Sulphur, LA 70663-6724
U.S.A.

FIRST CLASS MAIL

POSTMASTER: ADDRESS CORRECTION REQUESTED
RUSH -- TIME DATED MATERIAL

FIRST CLASS MAIL