COMING ATTRACTIONS

LA 99ers Toplcs

COMPUTER GROUP Newsletter

VOL 6 NO 4 APRIL 1987

PRESENTING

99'FEST-WEST'87

SECOND ANNUAL L.A. 99/4A EXPO
Hosted by
The Los Angeles 99er Computer Group
MAY 16th & 17th 1987
SHRINE EXPOSITION HALL
700 WEST 32nd STREET
Los Angeles, California
TWO FULL DAYS — 10 AM — 5 PM

PRESENTATIONS	MINI-WORKSHOPS
SEMINARS	NEW PRODUCTS
FAIRWARE	USER GROUPS
HARDWARE	SOFTWARE

ADMISSION $5 AT THE DOOR—GOOD BOTH DAYS

Three way hookup via GENIE connecting LONDON, OTTAWA and L.A.
* * * 10 A.M. SHARP - Saturday May 16th. * * *

***** SOCIAL EVENING with DINNER & DOOR PRIZES - MAY 16th *****
* ADVANCE RESERVATIONS NECESSARY *
* MEET THE MOVERS AND SHAKERS OF THE COMMUNITY *
* SEE THE LATEST INNOVATIONS IN HARDWARE AND SOFTWARE *
* For information on Hotel discount rates contact: *
* T.A.MASTERS, 148 S. MAPLE DRIVE, BEVERLY HILLS, CA. 90212 *
******* PLEASE INCLUDE A STAMPED SELF-ADDRESSED ENVELOPE *******

99'FEST-WEST'87 is in conjunction with Computer Sellathon & Expo

I n I

HI HO HI HO TIS OFF TO FAIRS WE GO

Hardly Snow White or the Seven Dwarfs, Tom, George and Terrie headed East for both the New Jersey and Massachusetts EVENTS. Communication and Camaraderie are the main benefits of this communion, another equally important is the learning process in observing the "Fests" of others. Our thanks for the opportunity to grow with the efforts of all.

Once again, the putting faces onto names was GREAT. Ellen Kramer, Henry Hein, Larry Hughes, Keith Koch, Ernest Chandler, Janet Ryan, Joyce Corker, Curtis Provance, R.A. Green, Helene LaBonville, among others are people we have long read and now met. True contributors to the community. We had previously met many of the other notables attendant at both locations, and it was Super to see them again. Really too many to mention.

New products and continuing support was very much in evidence. For an "ORPHAN" we certainly have a strong sense of survival. It has long been a pleasure to participate in the "brain-storming" sessions that occur when you put people like Tom and George in the same room with Barry Traver, Peter Hoddie, Howie Rosenberg, Paul Charlton or Chris Faherty. This time we had a "new" addition to this clan, Mike Dodd. Mike, his excellent programming ability and his enthusiasm, has been our mail and phone friend for some time. We knew he had a definite place in the community and communicated this to his Parents. They agreed and Mike was able to join us. It was a pleasure to meet and observe this young man, he instantly became a very active participant in the "brain-storm session". Mike previously wrote DM99 and his latest XBasher has just been released by Genial Computerware. He has just been elected President of the K-Town 99ers, having co-edited an excellent Newsletter for quite some time. Congratulations Mike.

There was an excellent answer at the FAYUH about the small and/or diminishing in number User Group. NUTMEG a group from Connecticut 12 members LARGE. What a dynamic bunch! They got together to jointly participate in an effort specifically for the Fayuh. A Fairware program called ZODIAC was developed. Each member took a month and Zodiac sign and researched notables born under that sign from 1930 to 1980. A formidable task! Material gathered, a data-base of information stored, excellent sprite graphics designed, and voila! a delightful, fun program created. This program can be easily used as a fund raiser for local group activity, $6.00 PP, Janet Ryan 10 Jolly Road Ellington Ct. 06029. Can you imagine what could be developed if only half of the still active User Groups were to make a similar joint effort. Don't lament CREATE.

Another example is Joyce Corker of Magnetic User Group, her TI WRITER TIPS AND TRICKS released by the Boston Computer Society is a run away success, $6.00 PP Boston Computer Society, TI-99/4A User Group, One Center Plaza, Boston Ma. 02108.

By the way, The Boston Computer Society is something to be envied!!! We were extremely impressed by its facilities. An umbrella for all Computer User Groups, it has just about everything one could want. Permanent location, extensive Library, various computers for use by members, Laser Printer, and on and on. Unique and wonderful.

New on the scene, the Gents from RAVE now have a speech card for the P-Box which will also be compatible with the 9640, to add to their product line. We also saw an alternative to the RAVE keyboard, by MLsystems - this one a well designed interface board containing 50 pre-programmed keys. Quite impressive and only $80.00, you can purchase any XT or PC keyboard at swap meets and off you go at a reasonable cost. You may buy this board from our Marketplace, as well as MLsystems, P.O. Box 268, Valley Falls, Rhode Island 02864.

Enhanced Display Package, Curtis Provance of Paragon Computing gave a very excellent demonstration of his outstanding program. It is available through our Library or directly from Curtis at 17 Constance Street, Merrimack, New Hampshire 03054, $10.00 Registration and Serial number, $15.00 with manual, $30.00 including source code. Both of the above will be demonstrated at a L.A. 99er User Group meeting "real soon now".

Canada, well the North American continent took a dip south with the additional weight of the number of our friends from up North supporting the 99/4A community, wow with their enthusiasm you just know their Fest on May 16 will be a roaring success, truly wish we could be in two places at one time. Ottawa, Good luck and a Great Fair, we will be chatting with you promptly 10:00AM Sat. May 16, on GEnie, along with our U.K. counterparts. Yes friends you do read right, there will be a 3-way Trans-World Conference between London, Ottawa and Los Angeles. If you are unable to attend one of the Fests on that day, do join us on the GEnie TI99/4A Roundtable. Thanks to GEnie for the cooperation in setting this up. Clive, Bob, et al, we will be typing it up with you all. And one more thing, Ottawa thank you very much, what a surprise.

Another strong supporter of our 99/4A community made the trek to Boston with part of her family in tow. Cheryl (Regena) Whitelaw, it was great to see you again, thanks for just being you.

99'FEST-WEST'87, yes we are really going to do it again, look for new participation here, RAVE, JOYPAINT, 9640, TURBO-XT, among others. Hopefully we will once again have Mike Dodd among us and have him demonstrate his great talent in person, J. Peter Hoddie, will you bring your Cello along please? Richard Mitchell will the NEW Handicap program be here? You know who is anxious to get it. So. California User Groups, the great success in Boston was the enthusiastic participation of North East area User Groups, we know we can do the same.

An open letter of thanks to, Sharon and Barry Traver, Joyce and Bill Corker, Walt Howe, Mike Wright, without all of you our trip East would not have been possible. Your generous and friendly hospitality was extraordinary, a thousand thanks.

Educators SIG, In Boston we had the distinct pleasure of meeting Jack Sughrue, professionally an Educator, and a staunch supporter of the 99/4A, it was brought up how necessary it was for Educators to have communication with one another. Bonnie Snyder also has similar qualities, it was suggested that we start off right here and turn the ball over to Bonnie and Jack, so if you teach, professionally or home school your children please communicate with Bonnie Snyder 62 S. Roosevelt St., Colorado Springs, Co. 80910, or Jack Sughrue, Box 459, E. Douglas, Ma. 01516. If you have teachers in your Group, please share this information with them. Thanks,

Finally CorComp, an attempt to reorganize out of chapter 11 is scheduled for April 23 1987, unfortunately the proposal offers very little or nothing to its creditors, among who are UPS, Federal Express, Pacific Bell, General Electric (just a few of the 85 creditors listed on exhibit B). There are 31 pages of documents, just too many to share, too bad.

GURU'S CORNER

by George F. Steffen

Recently, while looking at some assembly language programs which were of interest to me, I have seen some programming practices which, in contrast to normal usage, manage to both lengthen the program and decrease its speed.

```
LI  R0,>FFFF
```

There doesn't appear to be much wrong with that although, depending on the reason for loading that value, it might be more understandable to use -1 instead of >FFFF. The command uses only two words and will operate quite rapidly. However, there is a command which is better: SETO R0. This not only is shorter (one word) but operates faster. I mentioned this poor practice to Tom Freeman, and he told me of an even worse example.

```
ZERO  DATA 0
      MOV  @ZERO,R0
```

This example uses one word for data and a two word instruction in place of the one word command CLR R0.

Another routine which gives many programmers trouble is putting a number of identical bytes in VDP as in clearing the screen, for example. The routine I have seen most often is to set the VDP address in R0, the desired byte in R1 and the count in R2. Then do a BL to VSBW, increment R0, decrement R2 and loop till done. This is a very poor practice, since two address bytes are sent to VDP memory along with each DATA byte. A routine which moved two bytes at a time to R1 then used VSBW to move each to VDP was no better. VDP memory is self incrementing so that all we need to do is send the correct number of DATA bytes.

I have a routine which I wrote quite a while ago which I call VRBW (Video Repeat Byte Write) which uses the same conventions as TI routines in XB and EA. R0 is the VDP address, R1 is the byte to be written and R2 is the count. One additional subroutine address is needed. This is the routine used by VSBW and VMBW to put the address to be written in the VDP memory. I use VWAW (Video Write Address Write) as the mnemonic. Its address is >24CA in XB or >223A in EA or it is a simple task to copy the routine and inclue it in your program. My routine is as follows:

```
VRBW   BL   @VWAW     Get data for routine,set VDP addr.
VRBW1  MOVB R1,@VDPWD Send one byte to VDP
       DEC  R2        Count it
       JNE  VRBW1     Loop till done
       RTWP           Return to calling routine
```

This routine is a good compromise between speed and shortness. Speed could be increased by eliminating the loop and count and just repeating the MOVB instruction the correct number of times. This would increase length if there were more than two repeats. Another increase in speed could be obtained by loading R3 outside the loop with VDPWD and then changing the MOVB line to MOVB R1,*R3. This would shorten the loop by one byte by adding two bytes outside the loop. Program length would be one byte longer, but it would operate faster. I did not do this in my routine to retain compatibiliy with VSBW, VMBW, VSBR and VMBR. If you are using your own routines, the load R3 could be put in the VWAW (and VRAW—Video Read Address Write) routines (just one change necessary) and changing all five video read and write routines.

A practice which affects the program length in Source and Object code, but has no other effect on the program is using DATA or BYTE commands to give addresses of points within a block. For instance, pointing to the low bytes of R1 and R3:

```
MYWS  BSS  3
R1LB  BYTE 0,0,0,0
R3LB  BYTE 0
      DATA 0,0,0,0,0,0,0,0
```

This uses space for the BYTE and DATA statements as well as the LABELs. A better solution, which will use not use this space is:

```
MYWS  BSS  20
R1LB  EQU  MYWS+3
R3LB  EQU  MYWS+5
```

Of course, if you wish to preload certain information in these locations, you must use the first method.

Another time waster I have often seen is moving text for display on the screen one byte at a time and then adding the >60 offset before using VSBW to write it to VDP. Because the 9900 processor addresses only words but is capable of byte operations, it always reads a word before writing it, so that it does not destroy the other byte of the word. Therefore, to move a byte to a register, add the offset, and then move to VDP requires the following steps: Read source word, read register, combine the bytes, write source register, read offset, read source register, combine bytes, add offset, write source register and go to VSBW. It is much faster to have the text for display stored with the offset and then use the VMBW routine.

From the Disk of
Mike Dodd LA 99ers

GRAM Packer - V1.1... Peter Hoddie has released V1.1 of his excellent GRAM Packer program. V1.1 has many improvements over V1.0, including:

The part of the docs explaining the EA5 loaders are better written. Peter has included a "walk through" example that makes it far easier to understand.

GRAM Packer can now pack a loader for XBasic programs on the main menu. This has the disadvantage that it doesn't seem to work with my version of XBasic. I'm going to see Peter in April, so maybe we can figure out why then. However, don't let this deter you, as Peter says that I am the only one who has complained about it, which doesn't surprise me, as I am using my own modified version of XB that no one else has yet.

It can now pack a CALL on the main menu. For instance: the Horizon RAM-disk includes CALL DM to load Disk Manager 1000. Now you can pack a VERY short program that will execute a CALL DM. Note that this can not execute more complicated calls, such as CALL RUN("DSK1.UTIL1").

It can catalog a disk. When the program asks for a filename, type 1-4 for the catalog. GRAM Packer will display the name of each PROGRAM-type file on the disk. You can press space to get to the next one, or ENTER to select that file. Very useful if you forget the filename.

One of the problems with GRAM Packer was that it has to know whether or not it the program uses TI Save format. Now you can use an XB program I wrote to analyze a file and tell you what format it uses, whereas before, the only way to tell was trial and error. Since my program must read sectors off the disk, you must load Barry Traver's RAW program before running the XB program. RAW was on Genial TRAVeleR V1#4, and is present in all versions of the TRAVeleR's XXB program.

When the program runs, it will ask for a filename. It will then analyze the file and tell you if it uses TI Save, doesn't use TI Save, or if it isn't an EA5 file. It may take a while, depending on the number of files on the disk, since it is written in XB. The program is at the end of this article. Use Tom Freeman's Checksum program to key it in.

Peter Hoddie has just released (through Genial Computerware) XB:BUG, an Extended Basic debugger. XB:BUG made its debut at the March 28 TICOFF show.

XB:BUG is to Extended Basic what Super Bug II is to assembly language. With XB:BUG you can select a line (or range of lines) to breakpoint at, or you can hit SHIFT-CTRL at any time to call up XB:BUG. You then have several commands at your disposal, including:

Variable - list variables and their values

Change - allows you to change the value of variables. Very useful to have.

Array - inspect and change contents of array variables

Graphics - inspect character definitions, color table values, and the sprite table.

Data - shows which line the next READ command will read, and what it will read.

Files - shows the status of any OPEN file.

Program - shows the current line number being executed, the ON ERROR line number (if set), and the OPTION BASE.

Trace - a VERY powerful function. Trace will trace back all GOSUB and CALL sub's, back to the origin. This is on of the most useful commands you will find, I think. As an example: a program keeps entering a subprogram when it should not. Set a breakpoint at the start of the subprogram, and when it breaks, tell XB:BUG to trace back. Now you can find just where it entered the subprogram.

XB:BUG even has a calculator! Press +, -, /, or *, and XB:BUG will ask you for two numbers, and display the answer.

XB:BUG comes with five example programs, so you can quickly learn how to use XB:BUG. This is very helpful to have.

Note that XB:BUG will not work with MYARC XB II due to the completely different layout of memory. XB:BUG will work with any version of XB based on TI's XB, including Mechatronics XB II+ and MG's GK Extended Basic.

The docs included with XB:BUG are well written, and users should have no problems figuring out how to use it.

XB:BUG resides in low memory expansion, the area Extended Basic sets aside for assembly language programs. Therefore, there is NO limitation on the size of the XB programs. XB:BUG comes with a fast loader that loads it (about 5K) in seconds. If the XB program uses assembly subprograms, you can load a relocatable version of XB:BUG. It takes longer to load, but if the other assembly subprograms aren't too long, it should allow them both to fit. If the assembly subprograms ARE too long, you can load XB:BUG into high memory. That does have the disadvantage of requiring your XB program to have at least 6000 bytes of free memory at all times, or it will cause XB:BUG to crash.

For anyone who programs in Extended Basic, XB:BUG is invaluable. It is the ONLY program in its class. I think that you will find that is the best investment you could make toward programming in XBasic.

Final note: XB:BUG won overall prize in the Computer Shopper/TI Forum's first annual programming contest.

Micro Computing Journal: they finally paid... sort of. They sent me $21.75, after deducting $15 for MCJ V1. Ha, ha, ha. Surely they don't think that I'll be happy with that. I've written them a letter, and, as always, mailed a copy to the D.A. I'll let you know what happens.

Does anyone see some parallels between this and Star Wars? Lets see... I know! TI WARS: The Return of the

Users. In Return of the Jedi, the evil lord Darth Vader converted back over to the good side, so does that mean that Gary Kaplan will too?

TI-99/4A Linker... from R.A. Green, 1532 Chantenay Drive, Gloucester, Ontario Canada K1C 2K9. The Linker is a program designed to convert DIS/FIX 80 files over to E/A option 5 files, eaily and (relatively) painlessly. It can handle relocatable or non-relocatable files. To run linker, you generate a control file that consists of commands to process the DIS/FIX 80 file. With the control file, you can specify where the program is to be loaded, the entry point, patch the program, and more. Linker will also resolve all REF statements. It includes a library of all the subroutines in the Editor/Assembler, so that they are included in the program generated by the Linker. This way, the EA5 file generated by Linker can run out of ANY module, E/A or not. I haven't had time to run it on a lot of stuff, but it seems to work on almost every thing I've tried it on. There are a few files it won't handle correctly, though. I don't know why. TI-99/4A Linker is fairware. From what I have seen, it is well worth a donation.

```
100 !********************** !
245
110 !*GRAM Packer utility* !
168
120 !*Determines  if file* !
182
130 !*is TI-Save  type or* !
047
140 !*non - TI-Save  type* !
234
150 !*By Mike Dodd.  Uses* !
225
160 !*Barry Traver's  RAM* !
047
170 !*program.          * !
```

```
993
180 !********************** !
245
190 GOTO 200 :: A$,B$,C$,D$,
E$ :: A,B,C,D,E,F,G :: CALL
LINK :: !@P- !211
200 DISPLAY ERASE ALL:"MAKE
SURE BARRY TRAVER'S    RAM P
ROGRAM IS LOADED. IF   NOT,
PRESS FCTN 4 AND LOAD  IT" !
183
210 G=256 :: INPUT "FILENAME
? DSK":A$ :: A=VAL(SEG$(A$,1
,1)):: A$=SEG$(A$,3,10):: A$
=A$&RPT$(" ",10-LEN(A$))!216
```

```
220 CALL LINK("READ",A,1,B$,
C$):: B$=B$&SEG$(C$,1,127)::
 FOR E=0 TO 126 :: F=ASC(SEG
$(B$,E*2+1,1))*G+ASC(SEG$(B$
,E*2+2,1))!137
230 CALL LINK("READ",A,F,C$,
D$):: IF SEG$(C$,1,10)=A$ TH
EN 250 !224
240 NEXT E :: PRINT "ERROR -
 NOT FOUND" :: END !998
250 D=(15 AND ASC(SEG$(C$,30
,1)))*G+ASC(SEG$(C$,29,1))::
 CALL LINK("READ",A,D,D$,E$)
!192
260 E=ASC(SEG$(D$,1,1))*G+AS
C(SEG$(D$,2,1)):: IF E<>6553
5 AND E<>0 AND E<>887 THEN P
RINT "ERROR - NOT E/A 5 TYPE
 FILE" :: END !192
270 B=G*ASC(SEG$(D$,3,1))+AS
C(SEG$(D$,4,1))!112
280 E=ASC(SEG$(C$,17,1)):: F
=ASC(SEG$(C$,16,1)):: IF E=0
 THEN C=F*G ELSE C=F*G+E-G !
122
290 IF B=C THEN PRINT "TI SA
VE" ELSE PRINT "NON TI SAVE"
 !179
```

LI TopIcs

by Howie Rosenberg

I have been a member of the LA 99ers for quite some time now. Except for a few articles which have appeared in TOPICS, I am most likely unknown to most of you. You see it's somewhat difficult for me to attend meetings as I really can't seem to find the time to make the cross country trip that would be required to do so. I am one of a growing number of long distance members of your group. I live on Long Island, hence the name of this column.

Several others, also, will be contributing to TOPICS on a regular basis, I'm told. Barry Traver from Philadelphia and Mike Dodd from Tennessee will do so. I wish to personally thank Mike for last months contribution to TOPICS. His modification of VIDEO CHESS (Gram Kracker required), fulfills a prediction I made shortly after Gram Kracker was released. As a chess player the modification which allows saving and retrieving games to disk file makes VIDEO CHESS useful indeed! I hope to write in TOPICS on a regular basis (If Terrie and Tom let me). Right now in our orphan world the most important thing we can do, in my opinion is to communicate with each other. BBS', Magazines, and the network of newsletters of which LA TOPICS has been, for me tops. I have one comment on newsletters. One of the officers of a local group told me that their group did not distribute the newsletters which they received in exchange with other groups because the newsletters would get torn and dirty! Instead selected articles were copied in the groups newsletter for distribution to the membership. How sad! It would be far better for the exchange copies to be read, circulated, dirty, dog eared, and torn rather than crisp, new and unread. I suspect that a similar condition exists in many other groups. Come on guys! Let your members see what's doing in LA, Chicago, Colorado, and Washington. We are a closely knit community and let's talk to each other.

I expect that I will editorialize, sermonize, tutor, and write some code in this column. As I've spent a bit of time introducing myself this month, I'll limit myself to a short note in a series I started in Computer Shopper, Forth Bits. I expect that others in the BIT series will appear both here and possibly in Shopper. That's an advantage of these bits, independant (I hope) so they can be presented anywhere, in any order. Remember these articles are not intended to produce applications for you to type in but rather as tutorials in which the focus is on the process of developing code, algorithms and such.

FORTH BITS 5

Some Thoughts on the FORTH word BASE

One of the unique words in the FORTH language is BASE. Actually a user variable, BASE contains the number base presently in use. Storing a number in BASE makes that number the number base for keyboard entry and for display. Thus, entering 8 BASE ! would store 8 in the variable BASE. Entering the number 10 would now be equivalent to decimal 16. Two related words are HEX and DECIMAL. HEX is equivalent to 16 BASE ! and DECIMAL to 10 BASE !. I recall a recent article in, I believe, Smart Programmer in which a number of Decimal to HEX converter programs in XBASIC were compared. If memory serves me correctly, Barry Traver had an elegant, multi-statement one liner to solve the problem. How trivial it becomes in FORTH. : H/D DECIMAL . HEX ; defines a word H/D. Entering 10 H/D will return 16 and so FORTH. Programs to do arithmetic in several number bases become rather trivial. As an illustration, suppose I wanted to display the sum of two numbers in Decimal, Hexidecimal, and Binary. It's obvious before we write any code, that we will make repetitive use of the words HEX and DECIMAL. After you spend any time programming in FORTH it becomes equally apparent that it will make things both easier and neater if we define a new word : BINARY 2 BASE ! ; This last is a simple example of something, which if you read many of the FORTH bits, you will come to see as a subject I harp on as it is all important in writing readable, simple, FORTH code. Brody calls it factoring. When a sequence of FORTH words is used repetitively AND when that sequence can be described as a unique function, then and only then, should that group of WORDS be defined as a new word. As an example, if in looking through an application one found the expression DUP DUP + ROT a dozen times one would not define a new word using this sequence because it cannot be described as a function. Let's develop the idea further. First how are we going to execute the new word, lets call it 3BASE+ (I don't really like that name but we can't always be imaginative). Let's arbitrarily decide that we are going to start in decimal and will enter the two numbers on the stack followed by 3BASE+. Thus 3 7 3BASE+ will give us the sum of 3 and 7 in Decimal, Hex, and Binary. Entering + followed by the base desired and . will print the value of the numbers summed in the desired base and will also remove the two numbers from the stack. Thus we know that we will have to duplicate the two numbers on top of the stack for at least two out of three of the BASES. Of course, for the third base, (I hope I'm not beginning to sound like Abbott and Costello) we want to remove the numbers from the stack, another FORTH basic which cannot be repeated too often is WATCH THE STACK. Do not let it retain any uneeded data. An easy way to dup the top two numbers on the stack is with the word 2DUP which does not exist in TI FORTH but is created by : 2DUP OVER OVER ; (Appendix C of the TI FORTH manual). With the addition of a few format words and text the word 3BASE+ is now easily defined. : 3BASE+ CR 2DUP ." DEC " + . CR HEX 2DUP ." HEX " + . CR BINARY ." BIN " + . CR DECIMAL ; The last DECIMAL returns us to DECIMAL ready for the next conversion or whatever. Rather simpler to accomplish in FORTH than in other languages I would say!

Try the following. Change the number base to 64(64 BASE !). The base 64 has an interesting characteristic. The number set is represented by the full set of ASCII Characters. Thus HOWIE is a number (14 plus 18 times 64 etc). Of course the number HOWIE in base 64 is very large. It is too large to be represented by single length numbers. Thus if we type HOWIE . then HOWIE will not be returned. If instead we type HOWIE. A double length number will be stored. HOWIE. D. will return HOWIE. (end of sentence I did not want to put in two periods which would be grammatically correct). Most 5 character words fit into the set of double length numbers. Note of course that HOWIE. is not the same as Howie. (again another period) This characteristic of the base 64 is most useful for developing rather difficult to break cryptograms. The difficulty relative to normal letter substitutions is due to the fact that as we would use addition rather than letter substitution, the resultant carries make it quite difficult to solve by ordinary cryptographic means but that is the subject of another article. If one changes to base 36, than one is restricted to upper case letters and numbers but one can store 6 digit numbers. I have tried vainly to come up with pairs of numbers that give readable results ie HOWIE. SYLVIA. D+ D. TGAS0D. That sequence adds HOWIE (double length) and SYLVIA and prints the TGAS0D but alas it isn't. If someone who likes puzzles and is patient has some time, I would love to see some real sums of words(base 36 numbers which result in sense. Of course it is easy to get sensible conversions such as GOOD. 1000. D+ D. HOOD. or GOOD. 2. D+ D. GOOF but what I would like to see is two words summed to make a third. If it makes sense all the better!

One last thought relative to number bases (in a more serious vein). It has become customary when storing an application or set of words as a FORTH screen or set of screens, to store the number base that was in force prior to the LOAD and to restore it at the end. For this purpose the return stack is used and the code BASE->R is used at the beginning of the screen to store the number base on the return stack and R->BASE at the end to restore the previous base. Thus one can use any base one desires in the application and be assured that the original will be restored. Being creatures of habit most of us (including me) have used this technique and long ago forgotten what it really accomplished. If one examines the screens in the appendix in the TI FORTH manual one will find that the programmers who developed TI FORTH did likewise. When a number of screens are chained together in a single application there is no reason to use the technique on each screen. There is also no reason to use the technique when one is loading an application which will be used directly after loading (ie a program rather than utilities to be loaded as part of one's FORTH disk). It is easier to simply include a conversion to the required base at the end of the last screen loaded.

ASSEMBLY DISK CATALOG

by Tom Freeman

My subject for this month is the mysterious catalogging routine present in most storage devices. It should be instructive as to how this routine works, and also with regard to floating point numbers, which are not easy to understand.

Do you all remember typing in that "catalog" contained in the Disk Drive Manual? I remember actually believing at that time (in my primitive frame of mind) that there was a REAL file on the disk that contained the catalog, with real records etc. Boy did I get weird results when I tried to manipulate it in any way except the standard way that it was given to us. Well the reason is... Of course there is no file! What there IS, is a subprogram in the DSR of the card involved (floppy disk controller, ram disk, etc.) that is accessed when you open a dummy "file" that contains the device name and ends in period. What is returned to the PAB buffer with each record is a string and three numbers. THESE are in a standard format as if there were a real file. Each is preceded by a length byte (in the case of numbers, which are always stored in floating point format, this is 8). The first record contains the name of the directory or disk, a zero, the total number of sectors possible on the disk, and the number available. This information is contained in sector 0. The program actually has to compute the numbers. Each subsequent record will yield a file name, the file type (represented by a number from 1 to 5), the number of sectors used, and the logical record length. This information is obtained from the alphabetical list of pointers in sector 1, and each FDR it points to. My program takes advantage of this built in routine. I decided not to use direct sector access, because this requires a "drive number" and I wished to retain the ability to use device names not assigned to numbers (e.g. RD for Myarc RAMdisk). It was also meant to be able to catalog a hard disk, unfortunately this will require some additional code which I will publish within the next few months.

The second item of interest is "floating point numbers." These are always represented by 8 bytes. The number is expressedin RADIX 100 notation. Think of it as a *BASE 100 number. Each byte in the floating point* representation is a "place" which can be expressed,in decimal terms, by a single byte. For example the number 123 would have a "1" in the 2nd place to the left of the decimal, and a "23" in the 1st place. Places to the right of the decimal are expressed similarly. HOWEVER the 1st byte of the 8 is the "exponent" plus >40. E.g. the number 123 would have an exponent of 1,and thus would be represented as >41,>01,>17,... (the last 5 bytes don't matter). The 99/4A handles most "arithmetic" this way. There are a number of routines in the console that manipulate this type of number, as well as convert to ASCII etc. They are mostly accessed throught XMLLNK, as well as GPLLNK.

The program below is meant to be used out of command mode in XBASIC. It is not reliable from a running program because of its extensive use of VDP RAM , which is used for strings in running programs. There are a number of lines which can be eliminated if the program is to be attached to an assembly program, if that is the way you wish to use it. However you may have to "AORG" it so *as not to interfere with the rest of the program. It* works on an interrupt, always waiting for you to press CTRL C. When you do so, the screen clears and you are asked for the device name. After this is entered, you will get brief information on the screen as to the total volume, then nothing more until ALL th programs are read into memory. When this is done, the first page will flash onto the screen. You may then page up or down using the X and E keys. Paging is circular in both directions. Return to command mode is effected by pressing enter (which is also an escape from the input line).

Genial Computerware will shortly be publishing this program in an already assembled format, with several different AORG'd versions to fit with many differnt programs. The disk will also contain a patch for DISkASSEMBLER so that disks may be catalogged from any point within that program (this was done by popular request.

I hope you enjoy using and understanding this program.

```
       DEF  START,OFF,ON
PABLOC EQU  >1050
FAC    EQU  >834A
ARG    EQU  >835C
STATUS EQU  >837C
KEY    EQU  >8375
GPLWS  EQU  >83E0
VBF1   EQU  >1000
VBF2   EQU  >1080
VBF3   EQU  >1400
VBF4   EQU  >1800
*****************************************************
OFF    CLR  @>83C4       Clear ISR hook-turn off interpts
       RT
ON     LI   0,START      Load start of routine in ISR
       MOV  0,@>83C4
       RT
START  CB   @KEY,@CTRLC  Is CTRL C pressed?
       JEQ  S2           Yes, begin
       RT                No, return
S2     LWPI >83C0        Interrupt workspace
       MOV  13,@SAV13    R13-15 need to be saved for rtn
       MOV  14,@SAV14    -destroyed by FSUB routine below
       MOV  15,@SAV15
```

```
      CLR  @>83C4       Turn off interrupts temporarily
      LWPI WS           Our Workspace
* The next 8 lines are needed in Basic only, because all
* text must have basic bias of >60 added
      ABS  @BIASCK      Have we modified text already?
      JNE  S1A          Yes, skip next part
      SETO @BIASCK      Mark the change
      LI   0,DEV        Beginning of text to be changed
      LI   2,TYPES-DEV  Length (end-beginning)
S1    AB   @BIAS,*0+    Add the bias one byte at a time
      DEC  2            More?
      JNE  S1           Yes, go back
S1A   LI   0,>0285      Change screen image table from 0
      BLWP @VWTR        to >1400 (>400*5)-saves orig scr
      LI   0,>01F0      Text mode
      BLWP @VWTR
*A copy of value in VDP REG 1 (now in LSB of R0) must be
*placed @>83D4 because the value there is transverred to
*VDP REG 1 at every key press
      MOVB @WS+1,@>83D4
      LI   0,>0717      1=FG color,7=BG change if you
      BLWP @VWTR                          like
S3    LI   0,>0054      WRITE address for VDP >1400
      LI   2,>8C02      VDPWA                reversed
      MOVB 0,*2         Move LSB of >5400 first
      SWPB 0
      MOVB 0,*2         Now MSB
      DECT 2            VDPWD (>8C00) - as each byte is
*                       moved here, the address at >8C02
*                       auto-increments - Handy!
      LI   1,>1360      clear buffer(127 files bytes)
      LI   0,>0000      space with basic bias,use >2000
*                       if not in basic
S4    MOVB 0,*2         Because of auto-increment each
      DEC  1            byte written goes to next, with-
      JNE  S4           out changing R2
      LI   0,>142A      3rd Row, Col. 10
      LI   1,DEV        Text
      LI   2,9          Remember new screen image table
      BLWP @VMBW        write on screen
      AI   0,9          Prepare for input
      CLR  2            Counter
      LI   3,PROBUF+1   For storage
S5    BLWP @KSCAN       Look for key press
      MOVB @STATUS,1    Key pressed?
      JEQ  S5           No, go back
      MOVB @KEY,1       Yes put value in MSB of R1
      CB   1,@ENTER     Enter Key?
      JEQ  S6           Yes, process
      MOVB 1,*3+        Store value,increase buffer pos.
      AB   @BIAS,1      Add Basic Bias to R1
      INC  0            Next position
      BLWP @VSBW        Write on screen
      INC  2            Increase the counter
      JMP  S5           Go for more
S6    MOV  2,2          Enter pressed without text?
      JNE  S7           No, go on
      B    @ENDEND      Yes, branch to end
S7    BLWP @VSBR        Read the last character
      CB   1,@PERIOD    Is it a period?
      JNE  S3           No, go all the way back
      SWPB 2            Count in MSB of R2
      MOVB 2,@PROBUF    Put len byte at start of storage
      LI   0,VBF3       This is the screen image table
      LI   1,TIT1       Text
      LI   2,120        3 lines
      BLWP @VMBW        Write
      LI   0,PABLOC     Open mode
      LI   1,DSKPAB
      LI   2,10
      BLWP @VMBW        Write first part of PAB
      AI   0,9          Where len byte,dev.name will go
      LI   1,PROBUF     Where they are now
      MOVB *1,2         len byte into R2 (MSB)
      SRL  2,8          Now in LSB
      INC  2            Dev.name PLUS len byte
      BLWP @VMBW        Write rest of PAB
      MOV  @PABPT,@>8356 Pointer to location of len byte
      BLWP @DSRLNK      Open the file
      DATA 8
      JEQ  S3           Error, go back
      BL   @RECRD       "Read" 1st record (will contain
* disk name,then 3 #'s (0,total sectors,number available)
      BL   @CLRBUF      Clear buffer space
      LI   1,PROBUF     Where string will go
      BLWP @VMBR        Read string into it-see RECRD to
*                       see what R0,R2 have become
      A    2,0          Next item is a number(fltng pt)
      AI   0,19         The 1st # was 0(9 bytes),the 2nd
*                       we will use below(9 bytes)plus 1
*                       more byte to get to actual #
      LI   4,PROBUF+15
* Look at routine GETNUM to see what it does
      BL   @GETNUM      AVAIL in FAC & @PROBUF+15
      AI   0,-9         Now back to 1st #(TOTAL)
      LI   1,ARG        This will put TOTAL in ARG
      LI   2,8          AVAIL will still be in FAC
      BLWP @VMBR        both as floating point numbers
      LWPI >83E0
      LI   7,14         XMLLNK table is @>0CFA,FSUB is
      A    @>0CFA,7     7th entry (ARG-FAC)result in FAC
      MOV  *7,7         R7 contains address of routine
      BL   *7           Now USED in FAC
      LWPI WS
      LI   4,PROBUF+30
      BL   @GTNUM1      Convert to ASCII,put *R4
      BL   @ADD60       Add Basic Bias to the text
      LI   0,VBF3+11    Next three BLWP @VMBW instruc-
      LI   1,PROBUF     tions place the DISKNAME, AVAIL
      LI   2,10         and USED in proper locations on
      BLWP @VMBW        screen
      AI   0,15
      AI   1,16
      LI   2,4
      BLWP @VMBW
      AI   0,9
      AI   1,15
      BLWP @VMBW
```

```
      LI   9,VBF4       Initialize buffer to hold files
      CLR  @TOTAL       File counter
GETPRO BL  @CLRBUF      Clear PROBUF
* Each "record" will now produce a string which is the
* Filename, then 3 floating point numbers:1) file type
* negative if protected,2)size in sectors 3)record length
* if not "program"
      BL   @RECRD
      JEQ  END1         Null string=no more, jump to end
      LI   1,PROBUF+3   Read the name into PROBUF
      BLWP @VMBR
      A    2,0          Get to 1st number
      CLR  1
* 1st number is 1-5(+ or -),so 1st byte is ALWAYS >40
      INCT 0            therefore 2nd byte is IT
      BLWP @VSBR
      LI   3,>5920      "Y" or " "
      SRA  1,8          Number in LSB,but sign bit there
      JLT  GP1          If negative, leave R3 alone
      SWPB 3            Put " " in MSB of R3
GP1   ABS  1            Now get the positive # 1-5
      MOV  1,8          Save R1
      MOVB 3,@PROBUF+31 Put "y" or " " in proper loc.
      SLA  1,3          multiply by 8
      AI   1,TYPES-8    Index to file type
      MOV  1,3
      LI   4,PROBUF+19
      LI   5,7
      BL   @MOV34       Move the TYPE to PROBUF
      AI   0,8          Next number
      LI   4,PROBUF+13  Where to put
      BL   @SETNUM      Convert to ASCII
      CI   8,5          5="program", skip record length
      JEQ  GP2
      AI   0,9          Next number
      LI   4,PROBUF+25  Where to put
      BL   @SETNUM      Convert to ASCII
GP2   BL   @ADD60       Now add Basic Bias to all
      MOV  9,0          Location in VDP buffer
      LI   1,PROBUF
      LI   2,40
      BLWP @VMBW        Write it to the buffer
      A    2,9          Next position in buffer
      INC  @TOTAL       Counter
      JMP  GETPRO       Back for more
END1  LI   0,PABLOC
      LI   1,>0100      OPcode for close
      BLWP @VSBW
      MOV  @PABPT,@>8356
      BLWP @DSRLNK      Close the file
      DATA 8
      LI   4,21         Divide one R (4) into a contig-
      CLR  5            uous 2 word area(R5-6),integer
      MOV  @TOTAL,6     result in R5,remainder in R6
      DIV  4,5          R5=number of full pages of 21
      MOV  5,6                                  files
      MOV  6,@TOTAL     Now TOTAL=no. of full pages
* Note,we need one more "page" for end,but begin at 0!
      CLR  6            Initialize to first page
END2  LI   7,840        21 lines of 40 characters
      MOV  6,8          The page number
      MPY  7,8
* In the instruction MPY RX,RY the result will be con-
* in the 2 word sequence RY,RY+1 as a 32 bit number.What
* was in RY+1 before is wiped out.Thus in this case,R9
* will contain the result of the multiplication.
      AI   9,>1800      Now we know which page in buffer
      BL   @SCRO        Write to screen
END3  BLWP @KSCAN       Look for key press
      MOVB @STATUS,1
      JEQ  END3         None pressed
      MOVB @KEY,1       Move value of key press to R1
      CB   1,@ENTER     Is it "enter"?
      JEQ  ENDEND       Yes jump to end
      CB   1,@X         Is it "X"?
      JNE  END4         No, jump ahead
      DEC  6            Yes,down one page
      JLT  END5         Don't let it be <0
      JMP  END2         Go back and write new screen
END5  MOV  @TOTAL,6     Last page
      JMP  END2         And go back
END4  CB   1,@E         Is it "E"?
      JNE  END3         No,go back(no others allowed)
      INC  6            Up one page
      C    6,@TOTAL     Don't let it be >last page
      JLE  END2         OK,go back and write
      CLR  6            1st page
      JMP  END2         go back and write
ENDEND LI  0,>0200      Reset original screen image tabl
      BLWP @VWTR
      LI   0,>01E0      Reset 32 col.mode for Basic
      BLWP @VWTR
      MOVB @WS+1,@>83D4 Remember it needs to be saved
* Next 2 instructions return to original colors
      LI   0,>07F4              if needed (your choice)
      BLWP @VWTR        F4 are colors in GK XBASIC
      LI   0,START      Reload the ISR hook
      MOV  0,@>83C4
      CLR  0
      MOVB 0,@STATUS    Clear GPL status byte
      LWPI >83C0
      MOV  @SAV13,13    Restore the lost registers!
      MOV  @SAV14,14
      MOV  @SAV15,15
      RTWP              And back to basic!
**************************
* SUBROUTINES
**************************
* Clears PROBUF to spaces (without basic BIAS)
CLRBUF LI  3,>2020
      LI   4,PROBUF
      LI   5,20         20 words=40 bytes
CB    MOV  3,*4+
      DEC  5
      JNE  CB
      RT
* adds the basic BIAS to all 40 positions of PROBUF
ADD60 LI   1,PROBUF
```

```
      LI   2,40
A6    AB   @BIAS,*1+
      DEC  2
      JNE  A6
      RT
* Moves R5 bytes from R3 to R4
MOV34 MOVB *3+,*4+
      DEC  5
      JNE  MOV34
      RT
* Read a record,assume a string with length byte is
* first,get LEN into R2 and INC R0
RECRD LI   0,PABLOC
      LI   1,>0200      READ OP code
      BLWP @VSBW
      MOV  @PABPT,@>8356
      BLWP @DSRLNK
      DATA 8
      LI   0,VBF1       location of read buffer
      BLWP @VSBR        read 1st byte (length)
      INC  0            Beginning of string
      SRL  1,8          To LSB
      MOV  1,2          Put it in R2
      RT
* Sub to conerrt a floating point # at VDP addr in R0
* to ASCII and place it at addr. in R4,right justified
* GTNUM1 does same if # already in FAC
* The GPLLNK routine with DATA >14 takes a FP # in FAC
* converts it to ASCII starting at location pointed to
* by FAC+11 (must add >8300). If byte at FAC+11 is #
* then number will be in Basic format
* the length of the string is returned in FAC+12
GETNUM LI   1,FAC
      LI   2,8
      BLWP @VMBR        Put the Number in FAC
GTNUM1 MOVB 2,@FAC+11   MSB of R2=0
      MOV  11,10        Save return
      MOVB 2,@STATUS    Clear STATUS byte
      BLWP @GPLLNK      execute the routine
      DATA >14
      MOVB @FAC+11,3    Address of result,LSB
      SWPB 3
      MOVB @H83,3       MSB is >83
      MOVB @FAC+12,5    Length,includes leading space
      SRL  5,8          Put in LSB of R5
      LI   6,5
      S    5,6          This right justifies result
      A    6,4
      INC  3            These 3 instructions adjust for
      INC  4            the leading space
      DEC  5
      BL   @MOV34       Move to location in R4
      B                 Return
* MOVE 21 LINES FROM BIG BUFFER POINTED
* TO BY R9 TO VBF3
SCR0  LI   3,VBF3+120   Start on 4th line of screen
      LI   1,PROBUF     To transfer one line at a time
      LI   2,40         40 bytes
      LI   5,21         21 lines
SR1   MOV  9,0          Exact location in buffer
      BLWP @VMBR        Read to PROBUF
      MOV  3,0          Screen location
      BLWP @VMBW        Write on screen
      A    2,9          Change buffer location
      A    2,3          And screen location
      DEC  5            Go back for more
      JNE  SR1          If any!
      RT
*********************
* DATA AND BUFFERS
*********************
DSKPAB DATA >0000,VBF1,0,0,0  INTERNAL,RELATIVE,FIXED
DEV   TEXT 'DEV-NAME:'
*             0         1         2         3         4
TIT1  TEXT '  DISKNAME=           AVL=    USD=      '
TIT2  TEXT '    FILENAME  SIZE    TYPE    P         '
TIT3  TEXT '   ---------- ---- ---------- -         '
TYPES TEXT 'DIS/FIX DIS/VAR INT/FIX INT/VAR PROGRAM'
      EVEN
WS    BSS  32
PROBUF BSS 40
TOTAL BSS  2
SAV4  BSS  2
SAV11 BSS  2
SAV13 BSS  2
SAV14 BSS  2
SAV15 BSS  2
BIASCK DATA 0
D100  DATA 100
PABPT DATA >1059
CTRLC BYTE 131
READ  BYTE 2
CLOSE BYTE 1
H40   BYTE >40
H83   BYTE >83
ENTER BYTE 13
E     BYTE 'E'
PERIOD BYTE '.'+>60
X     BYTE 'X'
BIAS  BYTE >60
*********************************
* GPLLNK AND DSRLNK   MG VERSION,E/A GPLLNK won't work
*********************************************************
* Sorry, some of Craig's text is cut off!
GR4   EQU  GPLWS+8      GPL workspace R4
GR6   EQU  GPLWS+12     GPL workspace R6
STKPNT EQU >8373        GPL Stack pointer
LDGADD EQU >60          Load & Execute GROM address ent
XTAB27 EQU >200E        Low Mem XML table location 27
GETSTK EQU >166C
GPLLNK DATA GLNKWS  R7  Set up BLWP Vectors
       DATA GLINK1  R8
RTNAD  DATA XMLRTN  R9  Address where GPL XML returns t
GXMLAD DATA >176C   R10 GROM Address for GPL XML 0F 27
       DATA >50     R11 Initialized to >50 where PUTSTK
GLNKWS EQU $->18        GPLLNK's workspace of which
       BSS >08      R12-R15 registers R7 through R15 are
GLINK1 MOV *R11,@GR4    Put PUTSTK Address into R4 o
```

```
        MOV  *R14+,@GR6      Put GPL Routine Address in R
        MOV  @XTAB27,R12     Save the value at >200E
        MOV  R9,@XTAB27      Put XMLRTN Address into >200
        LWPI GPLWS           Load GPL WS
        BL   *R4             Save current Grom Addres on
        MOV  @GXMLAD,@>8302(R4) Push GPL XML Add on stack
        INCT @STKPNT         Adjust the stack pointer
        B    @LDGADD         Execute our GPL Routine
XMLRTN  MOV  @GETSTK,R4      Get GETSTK pointer
        BL   *R4             Restore GROM address off the
        LWPI GLNKWS          Load our WS
        MOV  R12,@XTAB27     Restore >200E
        RTWP                 All Done - Return to Caller
PUTSTK  EQU  >50           Push Grom Add to stack pointer
TYPE    EQU  >836D         DSRLNK Type byte for GPL DSLLNK
NAMLEN  EQU  >8356         Device name length pointer in V
VWA     EQU  >8C02         VDP Write Arress location
VRD     EQU  >8800         VDP Read Data byte location
GR4LB   EQU  >83E9         GPL Workspace R4 Lower byte
GSTAT   EQU  >837C         GPL Status byte location
DSRLNK  DATA DSRWS,DLINK1     Set BLWP Vectors
DSRWS   EQU  $                Start of DSRLNK workspace
DR3LB   EQU  $+7              R3 lower byte of DSRLNK work
DLINK1  MOV  R12,R12 R0  Have we already looked up the LI
        JNE  DLINK3  R1  YES! Skip look up routine
        LWPI GPLWS   R2,R3 Else load GPL workspace
        MOV  @PUTSTK,R4 R4,R5 Store current GROM address
        BL   *R4     R6
        LI   R4,>11  R7,R8 Load R4 with address of LINK r
        MOVB R4,@>4002(R13) R9,R10  Set up GROM with addr
        JMP  DLINK2  R11 Jump around R12-R15
        DATA 0       R12 contains >2000 flag when set
        DATA 0,0,0   R13-R15 contains WS, PC & ST for RT
DLINK2  MOVB @GR4LB,@>4002(R13) Finish setting up GROM add
        MOV  @GETSTK,R5 Take some time & set up GETSTK po
        MOVB *R13,@DSRAD1  Get the GPL DSR LINK vector
        INCT @DSRADD       Adjust it to get past GPL FETC
        BL   *R5           Restore the GROM Address off t
        LWPI DSRWS         Reload DSRLNK workspace
        LI   R12,>2000     Set flag to signify DSRLNK add
DLINK3  INC  R14           Adjust R14 to point to Callers
        MOVB *R14+,@TYPE   Move it into >836D for GPL DSR
        LI   3,>9F00
        MOVB 3,@>8400      TURN OFF SOUND GENERATOR
        MOV  @NAMLEN,R3    Save VDP address of Name Lengt
        AI   R3,-8         Adjust it to point to PAB Flag
        BLWP @GPLLNK       Execute DSR LINK
DSRADD  BYTE >03           High byte of GPL DSRLNK addres
DSRAD1  BYTE >00           Lower byte of GPL DSRLNK addre
        MOVB @DR3LB,@VWA   Set up LSB of VDP Add for Erro
        MOVB R3,@VWA       Set up MSB of VDP Add for Erro
        SZCB R12,R15       Clear EQ bit for Error Report
        MOVB @VRD,R3       Get PAB Error Flag
        SRL  R3,5          Adjust it to 0-7 error code
        MOVB R3,*R13       Put it into Callers R0 (msb)
        JNE  SETEQ         If its not zero set EQ bit
        COC  @GSTAT,R12    Else test CND bit for Link Err
        JNE  DSREND        No Error Just return
SETEQ   SOCB R12,R15       Error so set Callers EQ bit
DSREND  RTWP                  All Done - Return to
*************************
* VDP   UTILITIES    *
*************************
* modified by Tom Freeman to save a few bytes, and cor-
* for errors in case R0=0 in the multiple byte utilities
*************************
VSBW    DATA VDPWS,SBW
VMBW    DATA VDPWS,MBW
VSBR    DATA VDPWS,SBR
VMBR    DATA VDPWS,MBR
VWTR    DATA VDPWS,WTR
KSCAN   DATA VDPWS,KSC
VDPWS   EQU  $->14         20 BYTES NOT USED
        BSS  >C             ONLY NEED R10-15
SBW     BL   @WRITST
        MOVB @2(R13),@>8C00    MR1
        RTWP
MBW     BL   @WRITST
        JEQ  VRTN
MOREVW  MOVB *R10+,@>8C00
        DEC  R12
        JNE  MOREVW
VRTN    RTWP
SBR     BL   @READST
        MOVB @>8800,@2(R13)    MR1
        RTWP
MBR     BL   @READST
        JEQ  VRTN
MOREVR  MOVB @>8800,*R10+
        DEC  R12
        JNE  MOREVR
        RTWP
WRITST  LI   R10,>4000
        JMP  ADRESS
READST  CLR  R10
ADRESS  MOV  *R13,R12
        MOVB @VDPWS+25,@>8C02  LR2
        SOC  R10,R12
        MOVB R12,@>8C02
        MOV  @2(R13),R10       OLD R1
        MOV  @4(R13),R12       OLD R2
        RT
WTR     MOV  ,12
        MOVB @1(13),@>8C02
        ORI  12,>8000
        MOVB 12,@>8C02
        RTWP
KSC     LWPI >83E0
        MOV  11,@VDPWS+>16
        BL   @>000E
        LWPI VDPWS
        MOV  11,@>83F6
        LIST
        RTWP
        END
```

NOTE: Turn catalog on and off from Command level with CALL LINK("ON") and CALLLINK("OFF")

Did you know that...?

by Chick De Marti

THANX EUGENE 99/4A for these two items

If FCTN 4 is too much of a stretch sometimes, try this: (hold down at the same time) FCTN Space bar J - it works!

.....

Would you like to hook up an 80 column, display terminal to your TI and have it work too? Try hooking one up through the RS-232 Card - instructions are in the RS-232 manual. Then to access it through Extended Basic, type in:

```
LINPUT #1:A$
PRINT #1:A$
```

<*><*><*><*><*><*><*>

PUZZLE TIME

#1. You are camping and your camp leader (or wife) says,"I need exactly 6 gallons of water!" You. only have two containers. A 9 Gal. can and a 4 Gal. HOW ???

#2. With only 4 straight lines, • • •
(do not raise pencil off of
the paper) touch each dot. • • •

(Ans. next page - Chick) • • •

(The next two were found in the April issue of the **BAYOU BYTE**)

Here are two simple problems. One involving alot of logic and the other, a little math (and lots of logic).

(1) The Spider and The Fly
A 12x30 foot room has a 12 foot cealing. In the middle of the end wall, a foot above the floor, is a hungry spider. There is a fly in the middle of the opposite wall, one foot below the cealing. What is the shortest path the spider can take to get the fly? Give the distance in feet. (NOTE: This is a common spider; not able to fly or leap 30 foot distances).

(2) Time and Tide
A ship is at anchor in the harbor. Over it's side hangs a rope ladder with rungs one foot apart. The tide rises in the harbor at the rate of 8 inches per hour. At the end of 6 hours, how much of the rope will remain above water, assuming that 8 feet were above water when the tide began to rise?

See, I told you they were simple!

LIKE SHORT PROGRAMS?

PERMANENT SCREEN COLOR CHANGE
Found in the TI*MES

```
100 CALL CLEAR
110 B=2 :: F=16 ! your choice
120 C=16*(F-1)+(B-1)
130 CALL INIT :: CALL LOAD(9
984,C,C,C,C,C,C,C,C,2,0,7,15
+B,4,32,32)
140 CALL LOAD(9999,48,2,0,8,
0,2,1,39,0,2,2,0,8,4,32,32,3
6,2,0,8,8,4)
150 CALL LOAD(10021,32,32,36
,2,0,8,16,4,32,32,36,2,0,8,2
4,4,32,32,36,4,91)
160 CALL LOAD(-31804,39,8)
170 CALL LOAD(-31952,255,231
,255,231)
180 END
```

Set <B>ackground and <F>oreground colors of your choice (line 110)and save as "LOAD" on your disk.

<*><*><*><*><*><*><*>

TRACE SUBROUTINE by Mike Slattery
(reprinted courtesy TISHUG)

The next program will "take the TRACE line numbers off the screen and dump them to your printer...type it in at the end of your programand insert a gosub to the line number..."

```
9100 OPEN #1:"PIO"
9110 PR$=""
9120 FOR R=1 TO 24 :: FOR C=3
 TO 28 :: CALL GCHAR(R,C,X)
:: IF X=60 THEN 140 :: IF X=
31 OR X=32 THEN 150 :: IF X=
62 THEN X=32
9130 PR$=PR$&CHR$(X):: CT=CT+
1 :: IF CT>75 AND (X=32 OR X
=31)THEN PRINT #1:PR$ :: PR$
=="" :: CT=0
9140 NEXT C
9150 NEXT R
9160 PRINT #1:PR$ :: PR$="" :
: CT=0
9170 CLOSE #1: :: CALL CLEAR
9180 RETURN
```

(A GREAT debugging tool ... Chick)

(Did You Know ... cont.)

GENIE_BBS_HAS_FREE_UPLOADS

GENIE by General Electric is a modem seevice similar to Compuserve but much cheaper. Non-prome rate is $5 an hour for 300bd OR 1200bd! GENIE also turns off the clock when you upload a program! A TI Sig is goint stsrong along with a Krac Ker Box Sig for those with a Gram Kracker. For a FREE demonstration or to sign up call 1-800-638-8369 via MODEM only (use HALF DUPLEX). When connected type :
"HHH" and <E>nter. at U# type:
"5JM11961,GENIE" and hit <E> twice. The service will be explained to you with no obligation to join.

Thanx BOB DAGGITT and NINTY NINER NEWS.

<*><*><*><*><*><*><*>

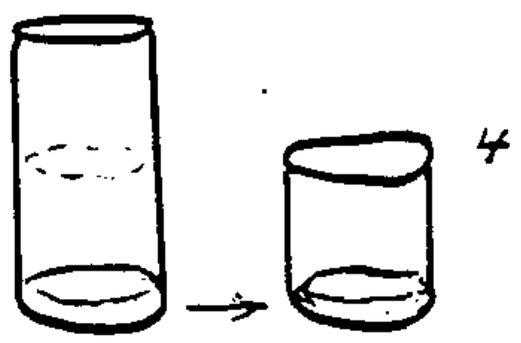
A TI-WRITER DOT

If you MUST have a dot in column one of your text, transliterate it. Using .TL try .TL 124:46 (FCTN A will now print a dot)

<*><*><*><*><*><*><*>

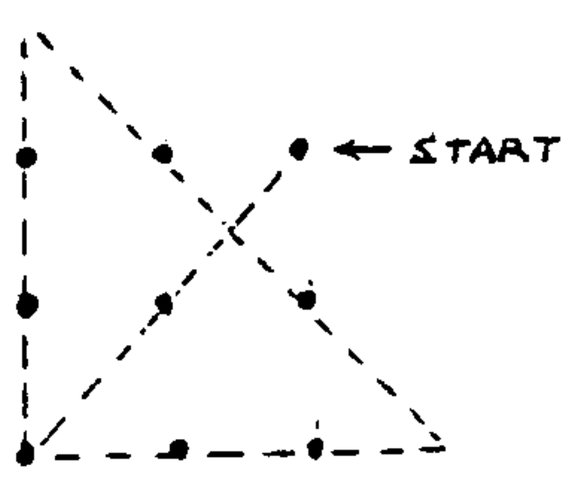
ANSWERS to problems #1 and #2

#1



Using the 4 gallon can as a measuring can, fill the 9 Gal can and transfer 4 gallons of water to the 4 Gal. can twice, dumping this out each time. You now have 1 gallon left in the 9 Gal. can. Save this in the 4 Gal. can. Fill the 9 Gal. can once more and fill the remaining (3 Gal.) space in the 4 Gal. can. You have the required SIX GALLONS OF WATER !

#2



(A lesson ... DON'T BE RESTRICTED BY ASSUMMED BOUNDRIES !)

DISK_SWEEPER2

by Steve Patterson
NEW HORIZENS

Steve's original DISK SWEEPER took 5 passes to delete all the files of a diskette. Thanks to many people around the U.S. who wrote articles on how to improve the program and Bill Sager, "who sent all the articles to me, I have a newer and better version...updated by
Adrian Robinson of California.

```
100 OPEN #1:"DSK1.",RELATIVE
,INTERNAL, INPUT
110 R=1
120 INPUT #1,REC R:A$
130 IF A$="" THEN 190
140 IF T<0 THEN 170
150 DELETE "DSK1."&A$
160 GOTO 120
170 R=R+1
180 GOTO 120
190 CLOSE #1
```

<*><*><*><*><*><*><*>

* * * From AKRON,OH - Thanx * * *

This discovery by one of their members about DM1000. To print a disk catalog of a different size:
1. at the first screen, hit FCTN # . enter your printer name (i.e. PIO) and control codes.
2. For 8 line/inch condenced print enter 27(space)31(space)1.
3. Save back to disk. To use, hit FCTN 7 at the Disk Utility Menu.

(NOTE: I haven't tried this, but it should work).

<*><*><*><*><*><*><*>

Out of coffee,
see you next month...Chick

KIDS ##########

```
100 CALL CLEAR :: CALL SCREE
N(6)
110 REM
120 REM ********************
        PGM BY SAM MOORE JR
        SHERMAN, TX 9/27/81
        ********************
130 REM
140 A$="<SPACE GEM>" :: FOR
GG=1 TO 7 :: DISPLAY AT(RND*
20,RND*20)BEEP:A$ :: NEXT GG
150 PRINT "DIRECTIONS? <Y/N>
"
160 CALL KEY(0,K,S)
170 IF S=0 THEN 160
180 IF K<>89 THEN 270
190 PRINT : :"THE OBJECT IS
TO MANEUVER   YOUR SPACE SHI
P TO AVOID    BEING HIT BY T
HE OTHER      SPACESHIPS."
200 PRINT :"A RUNNING TOTAL
IS KEPT OF  THE NUMBER OF TI
MES YOU ARE OVERRUN. THE OBJ
ECT, OF     COURSE IS TO MAK
E IT THROUGH"
210 PRINT "UNSCATHED. TO MAN
EUVER-ENTERS OR D OR E OR X
(ARROWS)."
220 PRINT : :"THE COMPUTER W
ILL ASK YOU   WHAT VELOCITY
YOU WANT.":"<2> IS A GOOD ST
ART."
230 PRINT : :"PRESS ANY KEY
TO CONTINUE..."
240 CALL KEY(0,K,S)
250 IF S=0 THEN 240
260 CALL CLEAR
270 PRINT "WHAT IS THE VELOC
ITY OF YOUR"
280 PRINT "SPACESHIP?(1-9)"
290 CALL KEY(0,K,S)
300 IF S=0 THEN 290
310 CALL CLEAR
320 V=K-48
330 V=V*10
340 PRINT "SKILL LEVEL DETER
MINES HOW  LONG THE GAME WIL
L RUN AND  SPEED OF THE ENEM
Y.":""
350 PRINT "WHAT SKILL LEVEL?
(1-9)"
360 CALL KEY(0,K,S)
370 IF S=0 THEN 360
380 LVL=K-48
390 CALL CLEAR :: CALL SCREE
N(4)
400 REM SPACE GEM
410 REM DEFINE SPACESHIPS
420 A$="0000070F107F7F10"
430 B$="0000E0F008FEFE08"
440 C$="0F070B112060F0F0"
450 D$="F0E0D08804060F0F"
460 CALL CHAR(104,A$)
470 CALL CHAR(106,B$)
480 CALL CHAR(105,C$)
490 CALL CHAR(107,D$)
500 CALL MAGNIFY(4)
510 REM MAKE SPACESHIPS
520 CALL SPRITE(#1,104,9,125
,100)
530 FOR AA=10 TO 15
540 SPEED=RND*LVL/5*60+RND*2
0
550 CALL SPRITE(#AA,104,16,1
,AA*45-445,SPEED,0):: NEXT A
A
560 CALL SCREEN(2)
570 REM MOVE RED SHIP
580 CALL KEY(0,K,S)
590 IF K<>68 THEN 600 :: CAL
L MOTION(#1,0,V):: GOTO 650
600 IF K<>83 THEN 610 :: CAL
L MOTION(#1,0,-V):: GOTO 650
610 IF K<>69 THEN 620 :: CAL
L MOTION(#1,-V,0):: GOTO 650
620 IF K<>88 THEN 630 :: CAL
L MOTION(#1,V,0):: GOTO 650
630 CALL MOTION(#1,0,0)
640 REM CHECK FOR HIT
650 CALL COINC(ALL,CC)
660 IF CC THEN 720
670 KK=KK+1
680 IF KK>29 THEN 810
690 MM=MM+1
700 IF MM=60+LVL*40 THEN 760
710 GOTO 560
720 CALL SCREEN(9)
730 HIT=HIT+1
740 FOR ZZ=1 TO 4 :: CALL SO
UND(-400,-5,5,ZZ*11+110,9,ZZ
*12+110,9):: NEXT ZZ
750 GOTO 560
760 REM END OF GAME
```

<SPACE GEM> continued...

```
770 CALL SCREEN(4):: PRINT "
END OF GAME": :"YOU SUFFERED
";HITS;"HITS"
780 PRINT : : : : : : :
790 FOR D=1 TO 999 :: NEXT D
800 END
810 REM CHANGE ENEMY MOTION
820 KK=KK-28
830 FOR AA=10 TO 15 :: SPEED
=RND*LVL/9+10
840 CALL SPRITE(#AA,104,16,1
,AA-455,SPEED,0)
850 NEXT AA :: GOTO 700
```

P A S S W O R D (by Chick De Marti

```
100 ! ************************
110 ! *  P A S S W O R D  *
120 ! * by Chick De Marti *
130 ! *     April 1987    *
140 ! ************************
150 CALL CLEAR :: CALL SCREEN(6)
160 FOR I=1 TO 8 :: CALL COLOR(I,16,1):: NEXT I
170 FOR I=9 TO 11 :: CALL COLOR(I,15,7):: NEXT I
180 SEC=23 :: W$="( warning )" :: CODE$="    LOOKPEEKDAREWAITLAFF"
190 DISPLAY AT(3,6):"P A S S W O R D": :TAB(9);"(       )"
195 DISPLAY AT(23,1):"( KEY WORDS ARE - HELP QUIT)":"LOOK PEEK DARE WAIT LAFF )"
:
200 D=800 :: GOSUB 1100 :: CALL HCHAR(23,1,32,64)
210 R=INT(RND*5+1)*4+1 :: M$="THIS IS STATION "&STR$(SEC):: GOSUB 1000 :: M$="EN
TER PASSWORD" :: GOSUB 1000
220 ACCEPT AT(12,3)SIZE(4):PW$ :: IF PW$="HELP" THEN 195 ELSE IF PW$="QUIT" THEN
 DISPLAY AT(20,8):"CREDITS=";CREDIT :: END
240 YE$=SEG$(CODE$,R,4):: DISPLAY AT(5,11)SIZE(-4):YE$ :: IF YE$=PW$ THEN 260
250 GOTO 300
260 M$="YOU ARE CLEARED....." :: GOSUB 1000
270 M$="100 CREDITS HAVE BEEN ADDED TO YOUR ACCOUNT" :: GOSUB 1000 :: CREDIT=CRE
DIT+100
280 IF CREDIT=500 THEN 500
290 SEC=SEC+5 :: M$="TRY TO ENTER SECTOR "&STR$(SEC):: GOSUB 1000 :: M$="PRESS S
ECRET KEYS" :: GOSUB 1000
295 ACCEPT AT(12,3)SIZE(4):Y$ :: GOTO 210
300 FOR I=1 TO 10 :: DISPLAY AT(16,9):W$ :: D=10 :: GOSUB 1100 :: CALL SOUND(-10
0,110,9,-3,9):: DISPLAY AT(16,9):"                " :: NEXT I
310 M$="BECAUSE OF YOUR ERROR" :: GOSUB 1000 :: M$="WE WILL DEDUCT" :: GOSUB 100
0 :: M$="25 CREDITS FROM YOUR ACCOUNT" :: GOSUB 1000
320 FOR I=10 TO 16 :: DISPLAY AT(1,1);"" :: NEXT I :: CREDIT=CREDIT-25 :: GOTO 2
10
1000 ! ** CRAWL **
1020 L=LEN(M$):: DISPLAY AT(10,1):"" :: D=5
1030 FOR X=1 TO L :: DISPLAY AT(10,X):SEG$(M$,X,1):: GOSUB 1100
1040 CALL SOUND(-100,1000,20,-4,5):: NEXT X :: D=30 :: GOSUB 1100
1060 RETURN
1100 FOR TIME=1 TO D :: NEXT TIME :: RETURN
```

DISK TO PRINTER PRINT

The January 1987 MICROpendium contained a program attributed to Bob Sims of the Nor-Cal TI users group. This program is called PRINTCOPY. It prints D/V80 files directly from disk to printer in an equivalent manner to TI Writer's editor. Changing the LINPUT in line 130 to INPUT will allow its use with internal type files.

The program requires Extended BASIC.

```
80 ! PRINTCOPY
90 ! BY BOB SIMS
100 INPUT "PRG NAME":PR$
110 OPEN #3: "PIO"
120 OPEN #1: PR$
130 LINPUT #1: A$
140 PRINT #3: A$
150 IF EOF(1) THEN 170
160 GOTO 130
170 CLOSE #1 :: CLOSE #3
```

TRITON SUPER EXTENDED BASIC MODULE ENHANCEMENTS AND ADDITIONS

MODIFIED COMMANDS

CALL VERSION	returns 120 instead of 110
CALL INIT	fixed bug in this call
LIST	user can specify column length 1-255
CALL LOAD	can now be used without memory expansion
PERMANENT	removed - never used
RESEQUENCE	removed - use RES instead
RES	allows RES for a block of program lines
TRACE	allows TRACE output to printer or disk

NEW FEATURES

ERROR MESSAGES	in upper and lower case
QUIT KEY CHANGE	quit key turned off
AUTO LOAD BYPASS	can bypass DSK1.LOAD with any key press
CURSOR MOVEMENT	program editing greatly enhanced with FCTN SHIFT and CTRL arrow keys

NEW COMMANDS

COPY	copy blocks of lines
DEL	delete blocks of lines
MOVE	move blocks of lines

NEW CALLS

CALL ALL(num var)	rapidly fills screen with char
CALL CAT("DSK1.")	catalog a disk
CALL CLOCK [1]	puts time on screen
CALL CLKOFF [1]	turns off clock
CALL CLSALL	closes ALL open files
CALL BEEP	beep sound
CALL HONK	honk sound
CALL CHIMES	chimes sound
CALL COLORS(f,b)	change all colors at once
CALL BYE	same as bye
CALL NEW	same as new
CALL GOSUB(num var)	allows numeric variable
CALL GOTO(num var)	allows numeric variable
CALL KEYS("keylist",num var)	allows valid key list
CALL PEEKG(addr,num vars)	peeks grom
CALL POKEG(addr,num vars)	pokes gram
CALL PEEKV(addr,num vars)	peeks VDP memory
CALL POKEV(addr,num vars)	pokes VDP memory
CALL QUITON	quit key on
CALL QUITOFF	quit key off - default
CALL STSPRT	stop sprite motion
CALL GOSPRT	start sprite motion
CALL SCRON	turns screen on
CALL SCROFF	turns screen off
CALL ALOCK(x)	checks alpha lock key
CALL SHIFT(x)	checks shift key
CALL CTRL(x)	checks ctrl key
CALL FCTN(x)	checks fctn key
CALL DRAW [1]	enables Draw and Plot

(see Draw and Plot manual for information on it's CALL LINKs, i.e. Edit, Show, Circle, Draw, Move, Fill, Plot, Gsave, Gload etc.)

1 - Memory Expansion Required.

NOTE: CALL DRAW and RUN A$ are currently being worked on and may or may not be included in the final module. All of the above enhancements are Gram Kracker Extended Basic enhancements except CALL DRAW. These enhancements were done by Danny Michael, Mike Dodd and Doug Warren.

NEW LA99 PROGRAMS FOR APRIL

2022 MATH ART #4 $5.00 E/A 4TH PROGRAM- Updated version- program will copy, has over 30 Graphics demo, fractals files, prints, clock, Etc. (SSSD)360

4028 MASS TRANSFER Version 4.1 $2.00 Freeware by Stuart Olson -A updated version. An Assembly Language Terminal Emulator, menu driven, X-Modem transfers, capable of multiple files transfers all at once. (SSSD)282

2021 SCREEN PAGER UTILITY $5.00 By Michael St.Vincent -Allow the viewing of listing from programs that are not currently in the memory. Basic or Extended Basic. Saves printed screen for instant recall, ideal for linking. Six programs. Instruction on printed material. (SSSD)45

4120 RE-DISK-IT $2.00 Freeware by John Schroeder X/B Will copy any disk SSSD, SSDD, DSSD, DSDD, Useing Corcomp or TI Controller. Requires memory expansion, two disk drives, X-B or TI-Writer or Editor/assembler or Gram Kracker (SSSD)67

4121 TI KEYS V3.0 $2.00 Freeware by Wes Johnson - Define 36 keys so that when typed as control keys, they will display up to 31 characters of text or code. Has preset commands if needed. (SSSD)146

4122 BACKUP V1.1 $2.00 Freeware by Michael Ballmann - Editor/Assemble (load and run)enter, TI-Controller only, Will Backup most disks on market. Gives Track #, side #, Sector #, Size of sector, Error code on read and write, Doc on disk, Source code available for $25 Doc on disk.(SSSD)62

4123 X/B HELPER $2.00 Freeware by Ken Houle - Short program to help you when you are in Extended Basic, Program is not yet complete. (SSSD)62

2737 MUSIC #37 $5.00 DIGITAL MUSIC BY STEPHEN D PEACOCK. PLAYS COMPOSE TEST SAVE MUSICAL SONGS. ASTON, BACH, BICYCLE, MARCH, HERO, HYMN, JOY, TAVERN, THANK - PLAY OR COMPOSE YOUR OUN MUSIC. (SSSD)306

2738 MUSIC #38 $5.00 Three great music programs- #1 JULY 4th Music and words with good graphic. #2 BUSY FINGERS amazing job of music writing. #3 TOCCATA plays Bachs and other with a nice X/B loader. (SSSD)329

2654 GAMES #54 $5.00 "MOONBASE" A very fast action shooting game written in Assembly. Great Color and graphic, Joy Stick, Use Editor Assembly Module #3 Load and Run MOONBASE enter and then PROC'D. (SSSD)220

2655 GAMES #44 $5.00 "MAJOR-TOM" Fabulous game for all. Written in Assembly use #3 MAJOR-TOM enter,PROC'D. Color and Graphic use Joy/stick or key board, Move Astronant thru maze. (SSSD)163

2428 EDUCATION #28 $5.00 10 Programs for the Astronomers. Trace the Comets, Planets, Moon, World latitude longitude and find out when the Sun raises and set on any date. ELLIPTICAL ORBITS, FULL MOON, WORLD MAP, SOLAR CALCULATOR, THE ASTRONOMER, PLANETARY ORBITS, HALLEY COMET, ASTRONOMER, ASTRONMY.(SSSD)255

2043 FAST SCREEN $5.00 By Bill Harms - Draw or create your own design on the screen by useing the Keyboard direction keys. Design can be printed. (SSSD)247

HOME HOME

2501 HOME #1 $5.00 15 programs for the home maker by AMNION- BUY OR SELL HOME, CREDIT CARD MANAGER, HONEY DO LIST, MAILING LIST, DIET MANAGER, KITCHEN AID #1, KITCHEN AID #2, EATS, LOAN COMPILER, 7 DAY CLOCK, AUTOMBILE COMPAIRSIONS, COOKIE FILE, THE FAMILY FOREST, HOME BUDGET MANAGER, HOME (SSSD)330

2502 HOME #2 $5.00 15 programs for the home by AMNION- HEAT AUDIT, COOL AUDIT, INCOME AVERAGING, PAYCHECK MANAGER, BILL TRACKER, HEALTH TEST, NUTRITIONAL SCOREBOARD, FAMILY TREE II, ANCESTORIAL FILE, HOME CHECK PROCESSOR, GAS AND ELECTRIC USAGE, PERSONAL FILE, RECORD COLLECTION, TRAINING CHART, RECIPE (SSSD)357

2503 HOME #3 $5.00 25 short programs for the home by AMNION- TV SCHEDULE, MESSAGES, HEATING, SAVING GAINS, CATALOG LIBRARY, FREEZER INVENTORY, WINE INVENTORY, PERPETUAL CALENDAR CLOCK, TRIPLE TIMER, ANNUAL BUDGET, HOUSEHOLD BUDGET AID, CHECKS BALANCE PRINTER, COOKIE FILE II, THE GROCERY LIST, BUDGET CONTROLLER, MENU PLANNER, ANTIFREEZE CONCENTRATION, MORTGAGE SPREADSHEET, HOME BUDGET SPREADSHEET, FURNITURE ARRANGER, DIET/RIGHT, FUEL COMPARATOR, SAVING ACCOUNT NUMBER, CHECKS SCANNER, DATE MINDER (SSSD)357

2504 HOME #4 $5.00 16 more programs to use around the home by AMNION- MENU MAKER, HOME ACCOUNT MANAGER, EXPENSE TRACKER, PERSONAL DIRECTORY, AMORTIZATION, HOME BANKER, BURGLAR ALARM, HOME FINANCE, UTILITY BILL ANALYSIS, DAMPER MAINTENANCE, AUTO LOANS, POISON PREVENTION, HOUSEHOLD INVENTORY, CUT CALCULATOR, HOUSEHOLD ACCOUNTS, MASTER INDEX (SSSD)341

2505 HOME #5 $5.00 13 programs for the homemaker by AMNION- LIST II, TAX ESTIMATOR, FINANCIAL CALCULATIONS, HOME BUDGET ANALYSIS, MAGAZINE FILE, BARTENDER, SOCIAL SECURITY, DAILY NUTRITION, INVESTMENT TRACKER, DEGREES DAYS, PERSONAL PROPERTY RECORD FILE, MAILING LIST, I.R.A. (SSSD)347

2506 HOME #6 $5.00 11 programs for the home by AMNION- BLACKJACK TUTOR, CREDIT UNION LOAN ACCOUNT, SELF EVALUTION. THE TAXKEEPER, IMPORTANT NUMBERS, CHECKBOOK ACCOUNTING TRANSACTIONS, LIST KEEPING III, SAVING AND LOANS ANALYSIS, QUILT PATCH, THE CLERK SYSTEM, WORD PROCESSING (SSSD)360

2507 HOME #7 $5.00 12 programs to use around the Home from LA99 Library - ADDRESS LETTER, CHECK PROCESSOR, FUZZY DECISIONS, GROCERY LIST, HOUSEHOLD BUDGET, INVESTMENT, LIBRARY BOOKS, PHONE BOOK, POWER GAS COSTS, RECORDS AND LISTS, TELEPHONE DIRECTORY, WORLD MAP (SSSD)329

2508 HOME #8 $5.00 14 record keeping programs from AMNION about Football, baseball, Basketball and Golf - GOLF HANDICAPPING, FOOTBALL RATING SYSTEM, BASKETBALL STATS, BASKETBALL STATS PART II, GOLF SCORE RECORDER, BASKETBALL STATISTICS, FOOTBALL STATISTICS, POWER RATER, NFL STATISTICS, BASEBALL STATISTICS, NAIA BASKETBALL STATS, GOLF SCORE ANALYSIS, FOOTBALL FORCAST, BASKETBALL STATISTICIAN. (SSSD)283

2513 ARTIFICIAL INTELLIGENT $5.00 4 kinds of artifical intelligent programs where the computer thinks for itself. DOCTOR, MINDREADER, DOGS, FETCH (SSSD)210

P8 I need the pins connections to connect a "ADAM" Printer to RS232 in my TI System. The ADAM Printer IS made by COLECO Model #72559 . If any of you out there know the pins connections PLEASE SEND it to me and I'll will make your life more enjoyable. I have no information on the Printer. **THANKS**

LIBRARIAN FRED MOORE 7730 EMERSON AVE. LOS ANGELES, CA 90045

XB: BUG
by J. Peter Hoddie

OVERALL WINNER OF THE FIRST ANNUAL TI FORUM AND COMPUTER SHOPPER PROGRAMMING CONTEST.

THE POWER OF XB: BUG:

- GIVES you the ability to look at the internals of your Extended BASIC program.
- CAN be called from a running program at a key stoke. Does not interfere with the program!
- PERMITS inspection of ALL Character, Color AND Sprite data.
- PERMITS inspection of ALL variables.
- PERMITS modification of numeric variables.
- ABILITY to list ALL subprograms.
- ABILITY to trace back ALL GOSUB's and subprogram CALL's.
- ABILITY to list program.
- ABILITY to SEARCH all graphics data, variable names, values and program listings.
- ABILITY to view all open files and their data buffers.
- ABILITY to view the next data statement.
- CHECK all system data, current line number, ON ERROR line, ON WARNING, ON BREAK, and more.
- CAN be used in conjunction with XB programs that use assembly language.

REQUIRES: TI Extended BASIC and 32K memory expansion.

The ULTIMATE tool for the Extended BASIC programmer. Complete documentation and several sample "debugs" included.

$15.00

TI WRITER TIPS AND TRICKS
By Joyce Corker

This supplement to the TI-Writer Manual will help you find answers to questions like:

- How did that happen?
- Is there a way to . . .?
- What did I do wrong?
- How can I do this quickly?

A must have for users of TI-Writer!

$5.00

HORIZON RAM DISK EPROM
by J. Peter Hoddie

THE RAM DISK EPROM:

- SUPPORTS single sided, double sided and 256K Horizon RAM disks.
- FASTER, more reliable than the original Horizon operating system.
- NEVER lose the operating system, again, because it's locked in ROM.
- ACCESS RAM disk as DSK1 to DSK6 and as HD.
- BUILT in CALL HDDIR to catalog RAM disk.
- CALL DM is available, PLUS, CALL EA5 for UTIL1, CALL MD for modem, CALL BOOT for BOOT program.
- SPECIAL BOOT program included, modified version of John Johnson's popular MENU program.
- CALL HDVOL to name RAM Disk.
- CALL HDDN to set drive number.
- E/A option 3 loader to load files from ANY device.
- ABILITY to change drive number at powerup.
- ABILITY to load BOOT, UTIL1, MD, or MGR at powerup.
- ALLOWS 14 extra sectors.
- BASED on the operating system the MYARC RAM Disk.

REQUIRES: Horizon RAM Disk. Included with EPROM are installation and modification instructions, manual, and disk.

$25.00

GRAM PACKER
By J. Peter Hoddie

THE MULTI--FACETED GRAM PACKER:

- ALLOWS you to store multiple EA5 programs in GRAM space for near instant access from main menu, CALL statements, or RUN command.
- SPECIAL utilities allow programs to be placed on menu, but to reside on disk, RAM disk, or hard disk.
- ALLOWS the running of Extended BASIC programs from main menu.
- ALLOWS for cartridge and even operating system loaders to be installed on menu.

REQUIRES: GRAM Kracker, GRAM Karte or Maximem. Complete documentation and disk included.

$10.00

"GRAM Packer is for everyone" — MICROpendium, December 1986.

MARKETPLACE

(the marketplace is a fund raiser for the club, that is, the "profit" goes to maintain the quality of this Newsletter. In general the price listed splits the difference between cost and retail. Please help your Club.)

MILLERS GRAPHICS	
DISkASSEMBLER	18.50
ORPHAN CHRONICLES (priceless)	9.95
ADVANCED DIAGNOSTICS	18.50
NIGHT MISSION	18.50
GK UTILITY I	10.00
SMART PROGRAMMING FOR SPRITES	6.25
NEW RELEASES	
UTILITIES DISK/DOCS (T FREEMAN)	8.00
PRE-SCAN IT! (J.PETER HODDIE)	10.00
GRAM PACKER "	10.00
FONT WRITER "	19.00
PRINTER'S APPRENTICE (M.McCANN)	19.00
MYARC	
RS232	82.00
D/D DISK CONTROLLER	155.00
128K RAM DISK/SPOOLER	175.00
512K RAM DISK/SPOOLER	280.00
EXTENDED BASIC II LEVEL IV	80.00
128K RAM DISK W/XBASIC II	235.00
512K RAM DISK W/XBASIC II	340.00
INSCEBOT	
TI-ARTIST	15.00
DISPLAY MASTER	12.00
ARTIST EXTRAS	6.00
GENIAL COMPUTERWARE	
XBasher (MIKE DODD)	9.00
XB:Bug (J.PETER HODDIE)	9.00
MEGATRONICS	
EXTENDED BASIC II PLUS	72.50
INTERN (BOOK ON GPL)	16.50
128K GRAM CARD	227.50
HARDWARE & SUPPLIES	
TEAC 55BV DSDD DRIVES	90.00
DISKETTES DSDD	.50
64K EPSON INT. PRINT BUFFER	45.00
COLOR RIBBONS (EPSON)	4.00
BACK ISSUES	
SUPER 99 MONTHLY	1.25
MICROPENDIUM	1.25
SMART PROGRAMMER JUNE 1986	1.50
BEST OF NEWSLETTERS W/DISK	5.00
FORTH NOTES VOL 1-6 (2.50 EA)	10.00
BEGINNER'S FORTH NOTEBOOK	2.50
ASSEMBLY NOTES VOL 1	2.50
TECHNICAL AND BUSINESS BOOKS	5.00
SAMS BOOKS (VARIOUS)	5.00
SAMS BOOKS WITH CASSETTES	7.50

(please send your order to the CLUB address, not the Librarian, and add $1.00 per disk for postage and handling. *CA residents add 6.5% tax*).

April 2, 1987

Dear Fairware Author,

May 16th and 17th, 1987 will mark the dates of 99'Fest-wesT'87. The event will be held at the Shrine Auditorium in Los Angeles, California, the site of 99'Fest-wesT'86. 99'Fest-wesT'87 will be sponsored by several 99/4A Users Groups in the Southern California area.

99'Fest-wesT'87 will be held in conjunction with the Computer Sellathon, as 99'Fest-wesT'86 was, and will feature education and information exchanges, commercial and user presentations, seminars, product displays of software and hardware, and a Fairware exchange booth, among many other surprise events. 99'Fest-wesT'86 was an exciting success, and we are all working even harder this year to make 99'Fest-wesT'87 bigger and better.

One area of great interest in the T.I. community is the Fairware concept. During 99'Fest-wesT'86, a Fairware booth was held to expose Fairware software and to obtain a fair contribution for the author's efforts. The success of the Fairware booth at 99'Fest-wesT'86 was due to the contributions of several Fairware authors including William Warren, author of PR-BASE, who could be found behind the Fairware booth on both days. A surprise visitor from Italy, Paolo Bagnaresi, was present with copies of his Fairware, BA-WRITER, also! Since most authors of Fairware have received little support from the 4/A community, it was proposed that Fairware could be distributed at 99'Fest-wesT'87 for a required donation and that most of the donation be forwarded directly to the author. The required donation will be based on the author's requested donation, or if no donation is specified, the donation will be set at $10.00. We will keep $2.50 to help defray the cost of disks, copying, postage for writing authors and sending them money, and to be of some help in defraying the 99'Fest-wesT'87 expenses. After the event, all authors receiving donations will receive the names and addresses of all who donated to their Fairware, as well as a check for the money due.

Because the 99'Fest-wesT'86 Fairware booth was so popular and provided the participating authors with a fair return for their efforts, we will again be offering a Fairware booth at 99'Fest-wesT'87 and would very much like your participation.

Most of the copies of Fairware for distribution at 99'Fest-wesT'87 will be obtained from the vast Fairware library of the L.A. 99ers Computer Group. As it is not always possible to keep up to date with all the latest versions of Fairware, if you would like to be sure that the most current version of your Fairware is distributed, please forward a copy to the address indicated below.

If for any reason you wish your Fairware not to be distributed at 99'Fest-wesT'87, please let me know.

Yours truly,

Steven D. Mehr

Steven D. Mehr
Fairware Coordinator
99'Fest-wesT'87

Steven D. Mehr
633 Hollyburne Lane
Thousand Oaks, Ca. 91360

ANNOUNCEMENT

March 19, 1987

Dear Readers,

It's 99'FEST-WEST time again! The dates for this year are May 16 & 17, 1987, again in conjunction with the Computer Sellathon at the Shrine Exposition Hall in Los Angeles. ALL OF OUR OUT-OF-TOWN READERS, AS WELL AS MEMBERS IN THE L.A. AREA ARE INVITED!!! Last year's FEST was quite successful, especially in light of the fact that it was our first venture into the TI Faire domain. We had many vendors and quite a respectable crowd, and most enjoyed the ability to see more computer products on the main floor of the Shrine.

We all had a rewarding and enjoyable time last year. Most of the out-of-towners stayed in the same motel and the hospitality room there was WELL used as many of our local members joined in on the fun, at ALL hours of the day.

Last year a discount coupon was distributed for a dollar off on the admission. This year the promoter has lowered the admission price for all, so instead we will be having a raffle for all TI ticket holders. The only way you can be identified as a TI user is to use the raffle ticket reproduced below at the time you purchase your ticket, which will be $5.00 for both days. As many copies may be made of this ticket as you wish.

If you have questions please call Terrie at 213-271-6930 or Tom at 213-454-1943, or write to us.

PLEASE COME AND ENJOY THE FUN!

Tom Freeman, President
Terrie Masters, Vice-President
LA 99'er Computer Group

TI 99/4A OWNERS
THE L.A.99ers and the USER GROUPS of the SOUTH WEST
PROUDLY ANNOUNCE
99/FEST-WEST/87
MAY 16&17, 1987 10A-5P
SHRINE EXPOSITION HALL
700 WEST 32nd STREET
LOS ANGELES, CALIFORNIA
$5.00 ADMISSION GOOD BOTH DAYS
VENDORS, HARDWARE, SOFTWARE, DEMONSTRATIONS, & MORE

PLEASE PRESENT THIS COUPON (REPRODUCTION ENCOURAGED)
FOR INCLUSION IN AN EXTENSIVE DOOR PRIZE RAFFLE !!!!
NAME_____

ADDRESS_____STATE___ZIP_____

TELEPHONE(___) _____ USER GROUP MEMBER?_____