Vol.3 No.4
APRIL 1985

THE SPRITE

a monthly newsletter of

THE 9900 USER'S GROUP, INC.

A voluntary organization for the sharing of knowledge and resources of people having interests in, or ownership of 9900 processor based Home Computers.

THE SPRITE is published monthly by **THE 9900 USER'S GROUP, INC.** for the enjoyment and furthering the knowledge of it's members in the use of 9900 processor based Home Computers. Address all correspondence to the EDITOR, THE 9900 USER'S GROUP, INC. P.O. Box K, Moorestown, N.J. 08057.

Articles from other newsletters are welcome and will be included to broaden our readerships base of knowledge and experience level. All submissions will be noted and credit given to the author. Articles from this newsletter may be reprinted for use by other user groups so long as the user group in question is an existing viable entity for the benefit of those wishing computer literacy. User groups that exist without a membership at large will not be considered user groups per-se. All articles or letters sent to the Editor for publication are subject to the unrestricted right to edit and comment.

THE SPRITE is available by subscription for $13.00 per year for 12 issues. Send *all* subscription payments and/or change of address to THE 9900 USER'S GROUP, INC., Subscription Service, P.O. Box K, Moorestown, N.J. 08057.

It will be the policy of THE 9900 USER'S GROUP, INC. not to pass, copy, or sell copyrighted text, cassettes, diskettes, or any other copyrighted medium thru The Group without the permission of the copyright owner.

Membership in THE 9900 USER'S GROUP, INC. does not impart to the general member any corporate authority or status to act for the corporation. Membership in THE 9900 USER'S GROUP, INC. is for one year from the month membership is acquired. Present membership rates are: $19/yr. as of January 29, 1985. Rates are subject to change without notice.

THE SPRITE is soliciting advertisers at the following rates:

FULL PAGE	$15.00	Note: All submitted
1/2 PAGE	$8.00	advertisements must
1/4 PAGE	$5.00	be printer ready.

We must receive all submissions by the 12th of each month for the next month's printing. Prepaid Orders Only!

THE GROUP OFFICERS and STAFF:

THE SPRITE STAFF

TI*BBS BULLETIN BOARD:

The Bulletin Board is available to all callers at no charge. Common courtesies prevail. The BBS is up most days 8AM - 11PM. The phone # for the BBS is 609-435-7301.

INTRODUCTION:

Where did spring go? It sure has been nice the last couple of weekends. Not much computing going on I'm afraid. Just sort of try to remember where you left your computer last. There must be something you can do with it. Let's see, tax time is over, but things like summer can put your computer to use. How about keeping track of the baseball teams statistics? Not the majors. Your kids! Although, for you serious baseball fans a serious listing of the league would put you right up on top come series time. Statistics do help to determine winners. There's the soccer teams too. How about pool maintenance? How about house maintenance? Let's face it. We all think of things we need to do to our homes but very often they don't get done. Why? Sometimes time conflicts, or money budgeting OR we forget. The computer doesn't forget. Why not use the computer to help you design that addition. A blueprint is a snap with our computrs. Layout your shed design with it. Using a 'smart' drafting program you won't be able to stick a 4ft 2x4 in that 3ft-11in space. The program won't let you forget about the thickness of the wood. Something easily overlooked when something is built WITHOUT a blueprint. How about dieting? MMmmm. Now that barbeques and outings are coming up I have some nerve mentioning calories huh. Well, there again. We may forget but the computer won't. So. Have fun this summer and let the computer do some of your work.

What else is new? Some of our members made it to the TRENTON STATE FAIRE and picked up some drives at reasonable prices. These shows are where you can get good bargains. Be carefull however. You may have to fix what you buy. If you do not have the expertise to do so the cost of repair may negate the savings. Always go to these shows with a keen eye and keep your cash in your pocket until the last moment.

DISKETTE SALES:

I'm going to try this again. We never seem to get enough folks who want to save money on diskettes. Why I don't know. Why in the world anyone would go and pay $29.95 or even $19.95 for a box of diskettes is beyond me. Here's the deal: $11.55 for 10 (reference only-read further) DSDD diskettes, not including sales tax. A LIFETIME GUARANTEE TOO! At these prices we don't get boxes. At these prices though who cares? You should have a diskette holder anyway. Here's the next catch. A MINIMUM of twenty diskettes AND I need at least 9 people. Notice that you DON'T have to buy in groups of 10!!! Just a MINIMUM of 20. You can buy 21 or 22 or whatever. If there is anyone who wants to buy 100 diskettes minimum I will drop the price to $11.30 per 10. On 100 diskettes that's a savings of $25.00!!!! If that occurs I will then only need 5 other people to fill the order. Are you catching on yet??? Two people who coordinate can order 200 diskettes plus at $1.13 EACH!!! Then I will fill the order. I will gladly take up the slack. I'm sure you know of some friends who need diskettes. FOR NON-MEMBERS please add .10 per diskette.

MINI-MEMORY CORRECTIONS: via CHUG newsletter

Here are some manual errors picked up along the way and passed along:

Page	HEX Line # as printed	Corrected Line #
60	)7E86 EC LI R0,)1300	EC LI R0,)1300
61	)7E8E PR MOV R11, R9	PR MOV R11.R9
	)7E90 AI R7,30	AI R7,30
	)7E94 LI R6,)6000	LI R6,)6000
62	)7EC4 AI R7,-32	LI R7,-32
	)7EC6 AI R4,-32	AI R4,-32

EOF For Fixed Length Files: by George Steffen via LA 99'ers

You may have noticed that the TI reference manuals suggest that you keep a note of the number of records in a fixed length file because EOF and APPEND statements do not work. However, with some modules that work with these files, using one record to keep the information is impossible. The following program will determine the last record in a fixed length file. You may then use that number in a RESTORE statement to add records, or use it in a counter to stop INPUTing records after the last one. Remember, the number of the last record is one less than the number found in this program, because the number of the first record is ZERO.

The program uses the CorComp Toolshed Utilities. It will run as is in Extended Basic. To run in Console BASIC, delete lines 100 and 110 and take out the subprogram name LINK. Then take the parentheses and quotation marks from around MPEEK and VPEEK. Of course, you must have the CorComp disk controller installed to use this program.

```
100 CALL INIT
110 DELETE "LD-CMDS"
120 PROGNAM$=RPT$(" ",10)
130 REM YOU MUST GET THE FILE NAME(INC-
   LUDING DISK NUMBER) IN THIS SECTION
140 OPEN #1:FILE$,FIXED,INPUT
150 FILE$=SEG$(FILE$&PROGNAM$,1,15)
160 CALL LINK("MPEEK")(33648,0,HI,LO)
170 LOADDR=256*HI+LO+11
180 FOR ADDR=LOADDR TO 15400 STEP 518
190 CALL LINK("VPEEK)(ADDR,0,DRN0,PROGNAM$,
   0,0,0,0,0,0,0,0,LO,HI)
```

```
200 LINE=ASC(PROGNAM$
210 LINE=LINE AND 127
220 PROGNAM$=CHR$(LINE)&SEG$(PROGNAM$,2,9)
230 IF (STR$(DRN0)=SEG$(FILE$,4,1))*
   (PROGNAM$=SEG$(FILE$,6,10))THEN 250
240 NEXT ADDR
250 LASTREC=HI6+LO
260 REM YOUR PROGRAM CONTINUES FROM HERE
```

PERSONAL RECORD KEEPING: by Newt Armstrong
via LA 99'ers

The TI99/4A is an enigma; many of its capabilities are only alluded to or are hidden. Take the Personal Record Keeping (PRK)* *module*, for example. Did you know that you can call seven PRK subprograms** from from TI BASIC if you have the module installed? Five of these allow you to create and access PRK formatted files, and the other two have the versatility of the ACCEPT AT and DISPLAY AT Extended Basic statements.

Now, you say, what earthly good does it do to put data from a Basic program into PRK format? Well, it allows you to massage your data with PRK, Statistics, and the Personal Report Generator modules. Best of all, to save your data on tape in 'program' format. How do you do it? Follow along:

PRK subprograms are named PREP, HEADER, GETPUT, LOAD, SAVE, ACCEPT, and DISPLAY. PREP is used to partition the Viseo Display Processor (VDP) RAM to provide a dedicated area for working on the PRK formatted file. HEADER is used to define the file structure and to retrieve housekeeping data for working on the file. GETPUT is used to transfer data between the file and the BASIC program. LOAD and SAVE are used to retrieve and store files in external storage devices. ACCEPT and DISPLAY accept data from the keyboard and displays it on the screen.

PREP: Prep is the subprogram invoked to partition VDP RAM for the work area. Format for the statement is:
CALL P(byte) - where byte is the number of bytes being reserved. The sequence is:

```
Main Title Screen
Press Any Key
Master Selection List
Press 1 for TI BASIC
Invoke CALL P(bytes) (enter)
(Disregard the next command if disk drive is
not connected)
Invoke CALL FILES(number) (enter)
Invoke NEW
```

The partition will remain in place until cancelled with BYE or the QUIT command. Size of the work area affects the amount of VDP RAM available for the basic program, as does buffer space. The CALL FILES(n) command reserves disk buffers; three are reserved automatically if the disk controller is connected to the console with power on. Each disk buffer uses about 520 bytes of RAM. An interesting exercise is to check memory available both before and after partitioning, and with one or more FILES called. You can use the following routine:

```
1 A=A+8
2 GOSUB 1
RUN
When the response is
* MEMORY FULL IN 1 *
Invoke PRINT A
```

Notice that preparation is in the Command mode and starts from console power on, essentially. If it is attempted with any basic commands in VDP RAM, an ILLEGAL CALL error will occur.

After the NEW command is invoked, the computer is ready to run a basic program. A PRK format file can be loaded into the work area from a storage device. Or one can be originated or manipulated with data from the basic program.

HEADER: Header is the subprogram invoked to define the file structure and to transfer housekeeping data between the file and the basic program. The header is page 0 of the PRK format file. The format for the statement is:
CALL H(n1,n2,n3,v($)) where n1 is the read/write code (1/0 respectively); n2 is the data code (1-14, see following list); n3 is the item number; and v($) is the data variable.

CODE	DATA	TYPE
1	File Name	0-9 characters
2	Day	integer (1-31)
3	Month	integer (1-12)
4	Year	integer (0-99)
5	Number of items per page (updated by routine)	
6	Number of pages (maintained by routine)	
7	Header length in bytes (maintained by routine)	
8	Page length in bytes (maintained by routine)	
9	Item Name	0-9 characters
10	Item Type: 1 = Characters 2 = Integers 3 = Decimal 4 = Scientific Notation	
11	Item Width: (Data window) 1-15 for Characters 1-10 for Integers 2-11 for Decimals	

	8-13 for Scientific Notation (maintained by routine)
12	Item decimal places 0 for Characters 1 to width-1 for decimal 0 to 5 for Scientific Notation
13	Item storage (bytes) (maintained by routine)
14	Item position in Page (maintained by routine)

Note that n3, the item number, is ignored for codes 1 thru 8 but must be included in the CALL statement as a space maintainer. Codes 9 thru 14 are repeated for each defined item.

As you can see, there is quite a bit of information to be included in the header. I think that it is easier to define file structure within the PRK program (I call it a Key File), and then enter and manipulate data from a basic program. Also, with data from codes 6, 7, and 8, you can determine the size of your file, and you will know how large a work area to allocate. PRK files are saved in 256-byte "chunks". So, the actual file length will be rounded to the next 256 multiple. TI, in the PRK manual, suggests a 2% overhead.

ED NOTE: If you exceed this safety margin and get the FULL warning it is TOO LATE!!! ALL IS LOST!! Beware of this pitfall.

GETPUT: Getput is the subprogram invoked to transfer data between the file and the basic program. Formats for the statement are:

CALL G(n1,n2,n3,v{$})

CALL G(n1,n2,n3,n4,v{$}) where n1 is the read/write code (0=write, 1=read, and 2=no data); n2 is the page number; n3 is the item number; n4 is the return code (used in the read statement only, 0=data found, 1=data missing); and v{$} is the data variable. Some what if's, must do's, and no-no's about the statement contents follows.

PAGES: Results are unpredictable for attempts to read from undefined, zero, or negative numbered pages. Pages should be created sequentially so numbers are not skipped. A page number in a write statement higher than any previously used will be the new highest page number stored in the header. An error will result from attempts to read a page numbered higher than the highest stored.

ITEMS: Items are defined with header write statements and are the same in all pages. Results are unpredictable for attempts to read from zero or negative numbered items. An error will result from attempts to read an item numbered higher than the highest defined.

VARIABLE: The variable must match data type (v for numeric, v$ for string) and item definition. When v is an expression, the evaluation will be written, and the evaluation must fit the item definition. e.g. An expression that results in a number with three decimal places will not fit in an item defined with two decimal places. Nor will an integer with four numbers or more (1000 up) fit an item defined to have a width of five with two decimal places.

LOAD: Load is the subprogram invoked to load a data file into the work area reserved by the PREP call. Format for the statement is:

CALL L(F$,n) where F$ is the file name ("CS1", "DSK1.___", etc) and n is a return variable. A return of 0 indicates an error occurred. Any other number indicates that the load was successful. Failures will be caused by a Call to a non-existing device or file., by general I/O errors, or by too small or no work area allocated.

SAVE: Save is the subprogram invoked to save a data file from the work area reserved by the PREP call. Format for the statement is the same as CALL L(F$,n)

ACCEPT: Accept is the subprogram invoked to receive data from the keyboard and to echo that data at a certain screen location. Formats for the statement are:

CALL A(n1,n2,n3,n4,v{$})

CALL A(n1,n2,n3,n4,n5)

CALL A(n1,n2,n3,n4,n5,n6) where n1 and n2 are row/column respectively. n3 ifs item width (data window); n4 is a return code (more about that later); v{$} is the item number, when it is the last numeric in the expression; or n5 is the low value of a low/high range with n6 the hi value.

As mentioned before, this statement is similar to to ACCEPT AT in extended basic. Data typed on the keyborad is accepted into variable v (for numeric) or v$ (for string) and is echoed on the screen starting at location n1(row), n2(column). Length of the input is governed by the value of n3 (data window) or the end of the row, whichever comes first. Say for instance, the input is ASHFOR, n2 is 23 and n3 is 8. Who wins? The row, in this case; the data stored and displayed is ASHFOR. Although the 8 space data window will accomodate the 7 letter input, there are only six spaces left on the row. The n4 return code allows processing of null entries and also for use of the function keys. Values returned for the various circumstances are listed below:

CODE	MEANING
1	Valid data entered
2	Empty (null) string
3	AID (F7) pressed
4	REDO (F8) pressed
5	PROC'D (F6) pressed
6	BEGIN F5) pressed
7	BACK (F9) pressed

When n5 is used alone, as the item number, the input will be checked for validity against characteristics stored in the header for that item. Invalid data (wrong type, too many decimal places, etc) will be greeted with the BEEP that we all recognize, and will be rejected. Using n5 in conjuction with n6 sets a range of valid data. Inputs outside that range will be rejected.

DISPLAY: Display is the subprogram invoked to write at a certain screen location. Formats for the statements are:

CALL D(n1,n2,n3,v($))

CALL D(n1,n2,n3,v1($),n4,n5,etc.) where n1 and n2 are row/column locations, respectively; n3 is item width (data window); and v($) is the data variable. Multiple displays can be made with one call listing several screen locations, data windows, and data variables in sequence. Length of this call is limited to the length of a basic statement. Positive valued data windows causes screen area clearing before data is displayed; negative valued windows leave area uncleared. As with the accept call, data that extends beyond the end of the row will be chopped.

EXAMPLES: Two sample programs are listed below. Prior to using them, you will have to prepare a header page (key file). Just go into the PRK module and define a file structure for six items- Last Name, First Name, Address, City, State, and Zip code. Save this information under some files name. Next, invoke CALL P(2000) as outlined above, and after the NEW statement, invoke CALL L to load your key file. Then run either the Read or Write program below, and happy computing. I am preparing a demonstration program to place in the library.

READ PROGRAM:

```
 10 REM READPRK/
 20 CALL CLEAR
 30 CALL SCREEN(13)
 40 CALL D(7,10,2,"1N",9,10,2,"fn",11,10,
  2,"ad",13,10,2,"ct",15,10,2,"st",17,10,2
 50 CALL H(1,6,0,RE)
 60 CALL H(1,5,0,FL)
 70 FOR R=1 TO RE
 80 FOR F=1 TO FL
 90 CALL G(1,R,F,MD,D$)
100 CALL D(5+2*F,13,12,D$)
110 NEXT
120 CALL KEY(0,K,S)
130 IF S()1 THEN 120
140 NEXT R
150 STOP
```

WRITE PROGRAM:

```
 10 REM SMPLPRKW/
 20 CALL CLEAR
 30 CALL H(1,6,0,R)
 40 CALL H(1,5,0,FL
 50 R=R+1
 60 CALL CLEAR
 70 CALL D(7,10,2,1n)
  9,10,2,"ct",15,10,2,"ad"
  13,10,2,"ct",15,10,2,"st",
  17,10,2,"zip"
 80 FOR F=1 TO FL-1
 90 CALL H 91,11,F,FW)
100 CALL A(5+2*F,10,FW.FR,D$)
110 CALL G(0,R,F,D4)
120 NEXT F
130 CALL KEY (0,K,S)
140 IF S()1 THEN
150 IF K=13 THEN 50
160 STEP
```

* (REFER TO PAGE 76, THE BEST OF 99'ER, Copyright 1983, Emerald Valley Publishing Co.)

** (I am indebted to my brother Al of the Southwestern 99'ers in Tuscon and to Jim Swedlow of the ROM staff for information about these subprograms, and to David Hough, also of the ROM staff, for some sample programs he 'just' happened to have in his library.

CORCOMP LOADER: via The 99'ers Assoc.

The following will load the CorComp Disk Manager from the Editor/Assembler module using option 3 - Load and Run.

```
     IDT   'LOADMNGR'   Program name
     AORG  )2700        Absolute load address
     DEF   MGR          Name in REF/DEF Table
MGR  LWPI  )83E0        Workspace Area & START Loc.
     MOV   R11,@)8300   Save return adx
     LI    R12,)1100    Load CRU Software base reg.
     SBO   0            Turn on Disk DSR
     SBZ   )000B        New Header value
     BL    @)44F2       Jmp to DSR start
     NOP                Delay
     SBZ   0            Turn off Disk DSR
     MOV   @)8300,R11   Restore old WS pointer
     B     *R11         Return
     END
```

TI DRASTIC PRICE REDUCTIONS: via NEW JUG NEWS

Just as a rememberence of things past the following is a list of DRASTIC PRICE REDUCTIONS as once announced by TI.

ITEM	PRICE	ITEM	PRICE
DISK DRIVE	$399.95	DISK CONTROLLER	$249.95
MULTIPLAN	$ 99.95	TI WRITER	$ 99.95
RS-232 CARD	$174.95	32K MEM CARD	$174.95

CORCOMP DSDD INITIALIZATION:

Why doesn't the CorComp card work right? My socks are too tight? My shoes are too big. Always questions. As you can tell only one question is pertinent. OR IS IT? I got together recently with another member and he brought over his DSDD drives and I had my old antiques (single sided) and we began..... I used the new Millers Graphics ADVANCED DIAGNOSTIC for the experiments. Ok. What we discovered was that depending on your drive, two factors are important with only ONE being predominant. We shall soon get to that.

First, let me clear up a BIG misconception. That of "double density drives." They exist ONLY in two places. Your mind, and the dollar signs running thru the 'sellers' eyes. All the work is done by the disk controller card. It runs the show and tells the drive at what density it is to write and also can set the Head Stepping Rate. Now on to the good news.

One factor that affected double density initialization was the Head Step time. Too fast and you got nowhere. There is a bottom limit to all drives. The initialization will occur at ANY step rate up to that bottom limit (usually around 6ms). At no time did we come up with a RANGE of values. Generally speaking 6ms (milli-seconds) is found to be the low limit. If you have brand new drives of recent production you may get them to work at 3ms but I guarantee you they had better be NEW.

The other factor is the MOST CRITICAL. If this is not right you can Head Step 'till the cows come home. We speak here of Motor Speed. The needs of some drives are more exacting than others. I kid you not. Sorry to say, on the most critical drives even the Advanced Diagnostics Motor Speed test is insufficient. That's not it's fault though. Always remember that a diagnostic 'tool' is not touted to be a corrective tool. It just 'diagnosis' (hence DIAGNOSTIC) the problem and then YOU have to fix it. On some drives, setting a larger number in the Motor Speed test of Advanced Diagnostic gives you a more accurate AVERAGE. It's STILL an average. If your drive fails to initialize after you have gotten the pointer right on zero (or close) without it moving then step two is in order.

Step two is the infamous 'flourescent lite' technique. This will get it set even closer! You of course need a flourescent lite and your drive turned upside down (most likely) or whatever you need to have the motor spindle staring at you. On the spindle will be those very same parallel dashed lines you see on your turntable and always wondered what they were. The flourescent lite by it's nature (you can't see it) fluctuates at a 60Hz rate. Like a stroboscope. That effect is used to 'strobe' those funny little lines. Now, if you shine your stroboscope at those lines they will 'appear' to stand still. If they DON'T, your drive is not at the proper speed. There are usually only two (or three at most) 'pots' (potentiometers or variable resistors) on the circuit board. They are easy to spot. Usually square or rectangular with a small screwdriver adjustment on them. Turn each one no more than one full turn (REMEMBER WHAT POSITION YOU WERE AT!!) until those lines either stop or change. If they do not change then you are at the wrong pot so put that pot BACK to it's original position and move to the next one. That's basically it. Remember, the drive must be running hence there's power. Don't short anything out or you'll be mad at yourself for a long time.

I started out with two drives that always gave me the infamous 'sector 7' error and then every 18 sectors after that and one drive that would occasionally give me errors. Immediately after adjusting the drive speed (on one drive I had to use the flourescent lite) my two external drives began to initialize double density, no errors. I have not had ANY problem since adjustment. The third drive is inside the p-box and will have to wait since I have to shut down and dismantle the whole system. But anyway, point proven. Adjust those drive speeds ANNUALLY or more often if you move and bounce your drives around.

MEETING AGENDA:

7:00PM - 7:30PM	Introduction, words of wisdom and whit.
7:30PM - 8:15PM	MULTIPLAN demo
8:15PM - 9:00PM	Open session, walk around, look at set up systems, buy stuff etc.

MEETING DATES:

MONTH	GENERAL MEETING	SPLINTER MEETING
APR	---) 29 (--note change	past
MAY	29	14 * NOTE

Plans are in progress to move us to Cinnaminson for the summer months. Dates and times to be fortncoming. Since Cinaminson requires a fee it is anticipated at this point that there will be NO combined splinter group meetings during our summer stay at Cinnaminson. If the separate splinter group meeting is still wanted by those who do attend then a minimal fee (we were supposed to do that anyway, remember) will be necessary.

*NOTE: The BASIC splinter group will meet on 13 MAY (Monday) at 7PM at the McDonalds meeting room <u>near</u> (NOT, in) Deptford MALL.

TIPS FROM THE TIGERCUB

#20

TIGERCUB SOFTWARE
156 Collingwood Ave.
Columbus, OH 43213

The entire contents of Tips from the Tigercub Nos. 1 through 14, with more added, are now available as a full disk of 50 programs, routines and files for just $15.00 postpaid!

Nuts & Bolts is a diskfull of 100 (that's right, 100!) XBasic utility subprograms in MERGE format, ready for you to merge into your own programs. Contents include 13 type fonts, 14 text display routines, 12 sorts and shuffles, 9 data saving and reading routines, 9 wipes, 8 pauses, 6 music, 2 protection, etc., and now also a tutorial on using subprograms, all for just $19.95 postpaid!

And I have about 140 other absolutely original programs in Basic and XBasic at only $3.00 each!(plus $1.50 per order for casette, packing and postage, or $3.00 for diskette, PPM) Some users groups charge their members that much for public domain programs! I will send you my descriptive catalog for a dollar, which you can then deduct from your first order.

Come on now, folks, don't you support your local schools? And don't you support those who support you? There are thousands of schools which have TI-99/4A computers in the classroom, usually without disk drive and without Extended Basic. They could use some educational programs in Basic on casette. They could probably use some of the public domain software in your library. Maybe they could use some of the educational programs I sell for just $3 (and I authorize schools to copy them for use within the school). There is probably such a school in your area - is your group supporting it? In the last Tips, I asked the members of 101 users groups to give me the addresses of schools that had TIs, so I could send them a free catalog. How many addresses did I get? Zero to the power of zero times zero!

More on the pestiferous asterisk bug in TI-Writer. Dr. Guy-Stefan Romano has confirmed and explained it. If you are printing out of the Formatter mode and your text contains an asterisk followed by two or more numeric digits - the asterisk and two digits will disappear! For instance, A*256 becomes A6, and I've noticed that A6 in programs published in several newsletters recently.

The TI-Writer program misinterprets the asterisk and two digits as an instruction to input data from a "value file" (see Alternate Input on p. 111 of the manual).

The solution to this bug is to type two asterisks followed by two dummy digits, then the actual digits. For instance, instead of A*256 type A**25256. Trouble is, the bug usually shows up in a program which has been LISTed to disk and then MERGEd into TI-Writer, and is usually not noticed. The solution? Run the program through my 28-Column Converter (see Tips #18!).

Dr. Romano informs me that there is an even worse bug in the Transliterate command coding, erratic and sometimes destructive. It is triggered by certain sequences of characters, but these have not been documented.

Dr. Romano says that he does not use transliteration.

I would suggest that you also avoid the use of the & and @. The & will only underline a single word, unless you tie words together with the ^ sign. If you tie words together, the Fill and Adjust will leave gaping blanks in your lines and if you tie too many together the line will extend beyond the right margin! Also, the underlining is a broken line. It is better to use the escape codes CTRL U, FCTN R, CTRL U, SHIFT -,CTRL U, SHIFT A, CTRL U, which will give a solid underline until you turn it off with CTRL U, FCTN R, CTRL U, SHIFT -, CTRL U, SHIFT @, CTRL U.

The @ is handy to emphasize a single word, but if you want to double-strike a whole sentence or paragraph it is better to use the escape code CTRL U, FCTN R, CTRL U, SHIFT G, and turn it off again with CTRL U, FCTN R, CTRL U, SHIFT H.

The period bug is another killer - the Formatter thinks that any line which begins with a period is a formatter command, and deletes the whole line! If your text contains a decimal value such as .11 and the wraparound puts it at the beginning of a line, the line disappears! There are two ways around this - put a 0 in front of all your decimals, as 0.11, or transliterate all your periods.

In all, the TI-Writer formatter is a temperamental and unpredictable piece of software, prone to unwanted line feeds and unexpected paper-wasting form feeds. I like to use it to right-justify text back to the disk, but from then on I prefer to print it out of the editor mode, or out of my own program.

Designing downloadable characters for the Gemini printer (see page 115 of the manual) is a bit tricky because it is hard to visualize how the expanded pattern will appear in print. The following program will enable you to experiment with designs, dump them directly to the printer for viewing, then save them as a file. When you later dump this file into printer RAM for use, you must activite the download characters with the escape code -
CHR$(27);CHR$(36);CHR$(1).

```
100 CALL CLEAR :: CALL SCREE
N(4):: CALL CHAR(128,"FF8181
818181B1FF",129,RPT$("F",16)
):: CALL COLOR(13,2,16)
110 FOR R=9 TO 15 :: CALL HC
HAR(R,11,128,9):: NEXT R
120 X=1 :: FOR R=9 TO 15 ::
DISPLAY AT(R,7)SIZE(2):STR$(
X):: X=X*2 :: NEXT R :: FOR
C=9 TO 17 :: DISPLAY AT(8,C)
SIZE(1):STR$(C-8):: NEXT C
130 DISPLAY AT(2,9):"TIGERCU
B'S" :: DISPLAY AT(4,1):"GEM
INI CHARACTER DOWNLOADER" !p
rogrammed by Jim Peterson fo
r the Public Domain
140 DISPLAY AT(17,1):" Move
cursor with W,E,R,S,D,":"Z,X
 and C keys. Toggle on":"and
 off with Q key. Press":"Ent
er when finished.": : :"Pres
```

```
s any key"
150 CALL KEY(0,K,ST):: IF ST
=0 THEN 150 :: CALL HCHAR(17
,1,32,224)
160 R=9 :: C=11 :: CH=128
170 CALL HCHAR(R,C,32):: CAL
L HCHAR(R,C,CH):: FOR D=1 TO
 10 :: NEXT D :: CALL KEY(3,
K,ST):: IF ST=0 THEN 170
180 ON POS("WERDCXZS"&CHR$(
13),CHR$(K),1)+1 GOTO 170,31
0,230,220,210,200,190,260,25
0,240,330
190 R=R+1
200 C=C+1 :: GOTO 270
210 C=C+1
220 R=R-1 :: GOTO 270
230 R=R-1
240 C=C-1 :: GOTO 270
250 C=C-1
260 R=R+1
270 R=R-(R<9)+(R>15):: C=C-(
C<11)+(C>19):: IF CH=128 THE
N 300 :: CALL GCHAR(R,C-1,GX
):: CALL GCHAR(R,C+1,GZ):: I
F (GX<>129)*(GZ<>129)THEN 30
0
280 DISPLAY AT(22,1):"You ca
n't have two in a row":"hori
zontally!" :: FOR D=1 TO 50
:: NEXT D :: DISPLAY AT(22,1
):" ":" "
290 CH=CH-1
300 CALL HCHAR(R,C,CH):: GOT
O 170
310 CH=CH+1+(CH=129)*2 :: IF
 CH=128 THEN 320 :: CALL GCH
AR(R,C-1,GX):: CALL GCHAR(R,
C+1,GZ):: IF (GX<>129)*(GZ<>
129)THEN 320 ELSE 280
320 CALL HCHAR(R,C,CH):: GOT
O 170
330 FOR C=11 TO 19 :: X=1 ::
 FOR R=9 TO 15 :: CALL GCHAR
(R,C,G)
340 IF G=129 THEN A=A+X
350 X=X*2 :: NEXT R
360 FOR J=1 TO LEN(STR$(A)):
: CALL VCHAR(15+J,C,ASC(SEG$
(STR$(A),J,1))):: NEXT J ::
M$=M$&CHR$(A):: A=0 :: NEXT
C :: A=0
370 DISPLAY AT(20,1):"Print?
 Y/N Y" :: ACCEPT AT(20,12)V
ALIDATE("YN")SIZE(-1):Q$ ::
IF Q$="N" THEN 470
380 IF F=1 THEN 390 :: F=1 :
: DISPLAY AT(20,1):"Printer
name?" :: ACCEPT AT(20,15):P
$ :: OPEN #1:P$
390 DISPLAY AT(20,1):"ASCII
to redefine?" :: ACCEPT AT(2
0,20)VALIDATE(DIGIT)SIZE(3):
CH
400 DISPLAY AT(20,1):"Descen
der (0 or 1)? 0" :: ACCEPT A
T(20,21)VALIDATE("01")SIZE(-
1):D$ :: D=VAL(D$)
410 M$=CHR$(27)&CHR$(42)&CHR
$(1)&CHR$(CH)&CHR$(D)&M$
420 PRINT #1:M$ :: PRINT #1:
CHR$(27);CHR$(36);CHR$(1);
430 PRINT #1:RPT$(CHR$(CH),7
2):: PRINT #1:CHR$(14);RPT$(
CHR$(CH),36)
440 DISPLAY AT(20,1):"Save (
Y/N)? Y" :: ACCEPT AT(20,13)
VALIDATE("YN")SIZE(-1):Q$ ::
 IF Q$="N" THEN 470
450 IF F3=1 THEN 460 :: F3=1
 :: DISPLAY AT(20,1):"Filena
me? DSK" :: ACCEPT AT(20,14)
:F$ :: OPEN #2:"DSK"&F$
460 PRINT #2:M$
470 M$="" :: DISPLAY AT(20,1
):"Another (Y/N)? Y" :: ACCE
PT AT(20,16)VALIDATE("YN")SI
ZE(-1):Q$ :: IF Q$="Y" THEN
100
480 CLOSE #1 :: CLOSE #2 ::
END
```

Micropendium ran a contest to improve on a brief ingenious organ program. The winner was Michael Christianson, who wrote a superb program. You'll have to buy the January issue of the magazine to get it (you should be subscribing, anyhow!). I didn't enter the contest, of course, and my version is not nearly as good, but have fun -

```
90 CALL CLEAR
95 PRINT TAB(5):"MICROPENDIU
M ORGAN": : : : : : : : :"Pl
ay bass with left hand": :"o
n left side of keyboard,": :
"melody on the right": : :
100 REM - MICROPENDIUM ORGAN
 modified by Jim Peterson
110 OPTION BASE 0
120 DIM NOTE(20)
130 FOR A=0 TO 20
140 READ NOTE(A)
150 NEXT A
160 DATA 40000,220,247,262,2
94,330,349,392,440,494,523,5
87,659,698,784,880,988,1047,
1175,1319,1397
170 CALL KEY(1,K1,S)
180 CALL KEY(2,K2,S)
190 CALL SOUND(-1000,NOTE(K2
+1),0,NOTE(K2+1)*1.01,5,NOTE
(K1+1)*3.75-ABS(K1+1=0)*1100
00,30,-4,0+ABS(K1+1=0)*30)
200 GOTO 170
```

A sprite routine that doesn't do anything but look pretty. I call it Patches.

```
50 CALL CLEAR :: CALL SCREEN
(5)
100 A$=RPT$("AA55",16):: B$=
RPT$("F",64):: CALL MAGNIFY(
4):: RANDOMIZE
110 FOR CH=40 TO 136 STEP 8
:: CALL CHAR(CH,A$,CH+4,B$):
: NEXT CH
120 C=2 :: S=40 :: R=1 :: FO
R T=1 TO 24 STEP 2 :: COL=15
0*RND+50 :: CALL SPRITE(#T,S
,C,R,COL,#T+1,S+4,C+1,R,COL)
:: S=S+8 :: C=C+1 :: R=R+15
:: NEXT T
140 FOR T=1 TO 50 :: CALL CO
LOR(#INT(24*RND+1),INT(16*RN
D+1)):: NEXT T :: GOTO 120
```

This is one that I fancied up, based on a sprite routine written by a youngster named Andrew Sorenson, published in the Sydney Newsdigest from Australia.

```
100 ! WILL O' WISP
      by Jim Peterson
       based on
       Andrew Sorensen's
       sprite routine
110 CALL CLEAR :: CALL SCREE
N(2):: CR=48
120 FOR CH=48 TO 63 :: FOR L
=1 TO 4 :: RANDOMIZE :: X=IN
T(16*RND+1)*2-1 :: X$=SEG$("
0018243C425A667E8199A5BDC30B
E7FF",X,2):: B$=B$&X$ :: C$=
X$&C$ :: NEXT L :: CALL CHAR
(CH,B$&C$):: B$,C$="" :: NEX
T CH
130 FOR N=1 TO 28 :: CALL SP
RITE(#N,CR,INT(14*RND+3),8*N
+20,120,5,0):: NEXT N :: IF
CR=64 THEN CR=48 :: T=T+1+(T
=2)*2 :: CALL MAGNIFY(T)
140 X=(INT(3*RND)-1)*4 :: Y=
(INT(3*RND)-1)*4
150 IF INT(10*RND+10)<>10 TH
EN 170
160 CR=CR+1 :: GOTO 130
170 FOR N=1 TO 28 :: CALL MO
TION(#N,-Y*20,X*20):: NEXT N
 :: GOTO 140
```

Here are a few more enhancements to my Menu Loader, published in Tips #15. Delete line 150 and add

```
101 OPTION BASE 1 :: DIM PG$
(127):: ON WARNING NEXT :: G
OTO 110
105 @,A,A$,B,C,D$,FLAG,I,J,K
,KD,KK,N$,NN,P$,PG$(),Q$,S,S
T,T$(),TT,VT,X
CALL INIT :: CALL LOAD :: CA
LL LINK :: CALL PEEK :: CALL
 KEY :: CALL SCREEN :: CALL
COLOR :: CALL CLEAR :: CALL
VCHAR :: CALL SOUND :: !@P-
```

The pre-scan will speed up run time by a worthwhile amount. The warning default will prevent a screen scroll on an erroneous Enter.

When you're finished printing strip labels, cut off the strip BEHIND the platen and roll it FORWARD! You'll waste a few labels that way, but if you try to roll backwards and get a gummy label stuck in the works, you've got trouble!

MEMORY FULL

Jim Peterson

the Tigercub

* Last Minute Page *

Computer Show & Flea Market

18 MAY 1985 at the George Washington Convention Center, Willow Grove, PA.
starts at 10AM
Exit #27 on Pennsylvania Turnpike

Through the group now available are the following schematics of the following cards:

RS232 card
Disk Controller card
P-code card
32k RAM card

Library Note:

Library is looking good. However, we need to get a good bulk order going to get disks so we can implement what we've got.

For right now, you will have to supply your own disks to the librarians. The same for cassettes right now.

T.L. Atkinson
28 Savona Ct
Dartmouth, Nova Scotia B2W 4R1, CANADA

CorComp, Inc.
1255 N. Tustin Ave
Anaheim, CA 92807

Extended Software, Co.
11987 Cedarcreek Dr.
Cincinati, Ohio 45240

Micropendium
Rt.2 Box 485
Round Rock, Texas 78644

Ronald L. Secord
904 Mayflower Ave.
Bloomington, IL 61701

Tigercub Software
156 Collingswood Ave.
Columbus, Ohio 43213

Home Computer Mag. User Grp Ed.
1500 Valley River Dr.,Ste250
Eugene, OR. 97401

The Suncoast Beeper Users Group
945 Montocello Blvd. N.
St. Petersburg, FL. 33703

Atlanta 99/4A Computer Users Group
PO Box 19841
Atlanta, GA 30325

Cin-Day Users Group
P.O. Box 519
West Chester, Ohio 45069-0519

COCHISE 99'er of Ariz. Users Group
172 James Drive
Sierra Vista, AZ 85635

N. Alabama 99 Comp. Users Group
PO Box 11204
Huntsville, AL 35805

Northern NJ 99/4A Users Group
PO Box 515
Bedminster, NJ 07921

Mid-Hudson 99'ers Users Group
PO Box 7298
Newburgh, NY 12550

M.I.T. Lincoln Lab. TI-994A Users Group
8820   90 Way North
Seminole, Florida 33543

North Jersey TI Users Group
21 Morse Ave
Butler, NJ 07405

Phila TIBBS Users Group
PO Box 2733
Philadelphia, PA 19120

Los Angeles 99'rs Users Group
PO Box 3547
Gardena, CA. 90247-7247

Capital Area Users Group
PO Box 637 Federal Sq. Sta.
Harrisburg, PA 17108-9998

Central Iowa 99/4A Users Group
P.O. Box 3043
Des Moines, IA 50316

Arizona 99 Users Group
4328 E. LaPuente Ave.
Phoenix, AZ 85044

Amarillo 99/4A Users Group
PO Box 8421
Amarillo, Texas 79114

Chattanooga 99/4A Users Group
P.O. Box 136
Hixson, Tenn 37343

Lehigh 99'er Computer Users Group
PO Box 4837 - 1501 Lehigh St
Allentown, PA. 18103

Delaware Valley Users Group
PO Box 6240
Stanton Branch,Wilmington,DE 19804

Aloha 99/4A Computer Users Group
92865 Palailai St
Makakilo, HI 96707

Wash. DC Area Users Group (TI NEWS)
Box 267
Leesburg, VA 22075

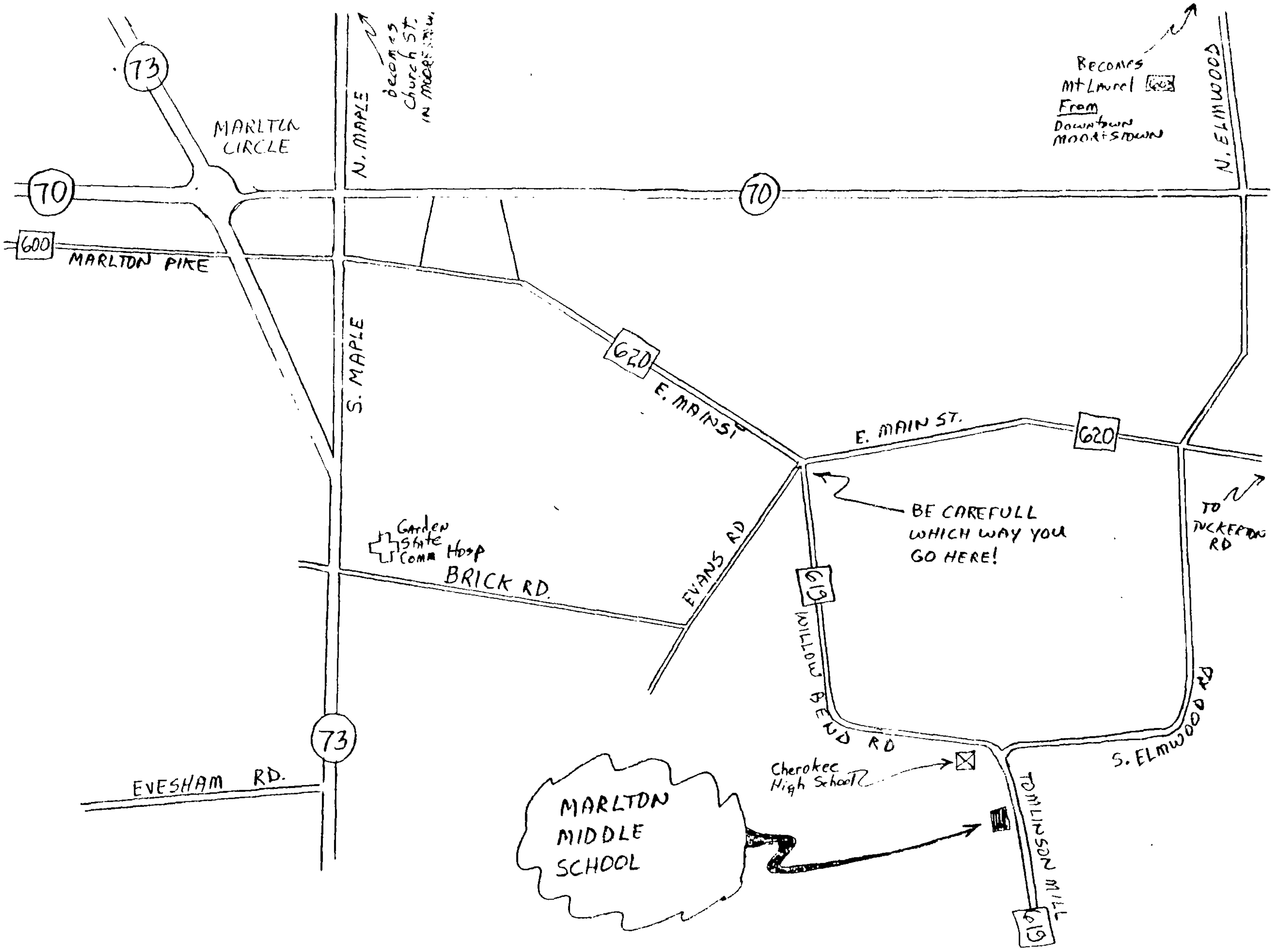
99'ers Users Group Assoc
3535 South H St. #93
Bakersfield, CA 93304

KENTUCKIANA 99/4 Comp. Users Group Soc.
9801 Tiverton Way
Louisville, KY 40222

The R O M Users Group of Orange Cnty
17301 Santa Isabel St.
Fountain Valley, CA 92708

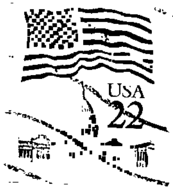
Edmonton 99'ers Users Society
PO Box 11983, Edmonton
Alberta, CANADA T5J-3L1

Wycove Systems Ltd.
PO Box 499
Dartmouth, Nova Scotia B2Y 3Y8 CANADA


73
MARLTON CIRCLE
70
600
MARLTON PIKE
N. MAPLE
becomes Church St. in Moorestown
Becomes Mt Laurel 603 From Downtown Moorestown
N. ELMWOOD
70
S. MAPLE
620
E. MAIN ST
E. MAIN ST.
620
BE CAREFULL WHICH WAY YOU GO HERE!
TO TUCKERTON RD
Garden State Comm Hosp
BRICK RD.
EVANS RD
619
WILLOW BEND RD
S. ELMWOOD RD
Cherokee High School
TOMLINSON MILL
619
MARLTON MIDDLE SCHOOL
73
EVESHAM RD.

THE 9900 USER'S GROUP, INC.
P.O. BOX K
MOORESTOWN, N.J. 08057

MSC. SO[illegible] JERSEY, NJ 0803
MAY 13 PM



Edmonton 99'ers Users Society Exchng
PO Box 11983, Edmonton
Alberta, CANADA T5J-3L1