

80 Plus Columns In TI-Writer!
by Mike Ewell

You can print past 80 columns in both the Editor and the Formatter. First you need to put your printer in condensed mode. In the editor you would do this in Special Character Mode by pressing Control U once, press Shift O once, and press Control U once more. Then do a Print File to PIO.CR or whatever your printer requires and it will print your file the full width of your printer. If you exceed your printers right margin before a carriage return is sent, your printer will chop off your text at that point.

In the formatter, you can either use the Special Character Mode or use a Transliteration Command such as: .TL X:27,15 where X equals the ASCII value of the item that you want to trigger condensed mode. You would set your right margin to 135 or whatever you need and use the Fill command. The hard part to printing this way is watching your right margin.

In a Print File, If you need to print at a specified point such as filling in a blank space on a form, you can either count your spaces or use the tab function of your printer. Press Control U once, press Shift I as many times as you need (my printer tabs every 8 columns) and press Control U once more. If you need to be between tabs then use spaces after your control code and before your text.

In the Formatter, you can either use this method or if your printer allows, reset the tabs with a TL command and then use a single control code to move your text around. Practice!

SBTIUG GENERAL MEETING 2 JUNE, 1988

by Nora Knudsen

The June meeting of the SBTIUG was held at the Saratoga Library on the first Thursday, June 2nd, with 15 members in attendance.

President Mike Ewell opened the meeting at 7:25 pm with a request for a report from Treasurer, Kevin Daberkow who stated that after receiving 2-3 renewals and disbursing approximately \$38.00 for expenses, the current balance stands at \$533.84.

Next was a report from Librarian, Helmut Fuchs. Helmut announced that version 3 and version 4 of C99 are now available in the club library.

Mike Ewell stated that he had passed the new BBS program to Keith Felix, who, although experiencing some modem difficulty, expects to have the BBS up and running soon. Don Apte's report as usual had lots of swaps and computer meets.

Bill Schult still needs articles for the newsletter!

Ray Keller commented on re-inking printer ribbons. Glen

Johnson said he tried it and thought it worked well.

A discussion ensued on preferences for future demonstrations. Multiplan seems to be the popular choice and no one seemed to mind starting at the beginning.

The General business meeting over, we adjourned for a demo on TI Writer by Mike Ewell.

EDITORS RAMBLINGS

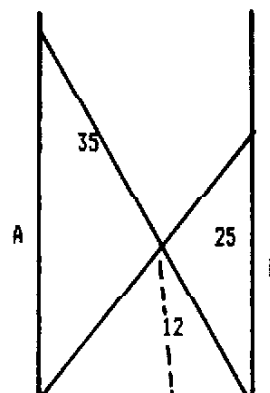
I wish to thank all those members that have contributed items and help for the newsletter. It has been greatly appreciated. We however do need more original articles. With the talent that we have in our group we should be able to come up with articles and programs that will be of interest to our group and the TI community in general.

Another item of concern is our dwindling membership. TI produced several million of the TI 99/4A computers. Many of these are in the Bay Area sitting in closets gathering dust. We need to find a way to interest the owners to retrieve these little gems from their hiding places and reactivate the owners interest in their computer. We further need to find some way to show these people how our users group can be of a help to them.

I still have a need to know what the general membership of the group wants to see in their newsletter. I receive a large number of newsletters from other Users Groups and extract articles from them for use in our newsletter. So if you have a particular interest, let me know at the meeting and I will include articles that interest you in our newsletter.

CHALLENGE CHALLENGE CHALLENGE

Consider an old problem found in many classic collections. Two ladders one 25 feet long and the other 35 feet long lean against sides of buildings on opposite sides of an alley as shown below. The point at which the ladders cross is 12 feet above the ground. Write a program to determine the width of the alley.



LET'S TALK RAM DISKS PART 4
by John F. Willforth
reprinted from WEST PENN 99'er

The MYARC RAM DISK is one of the most popular and versatile units on the market. My thanks to Scott Coleman for his input to the preparation of this article. Scott has the 512K version, and this is the model that most of you would probably migrate to in time.

The MEMORY EXPANSION CARD (as MYARC calls it), comes in three sizes, the basic 32K unit, the 128K model, and the 512K RAM DISK. The card is architecturally similar to the 128K card from FOUNDATION, in that it uses 32K RAM space. It has up to 16 banks of 32K, with the entire 32K being switched at once, verses the 2K bank switching occurring in the MRD.

The MYARC unit is supported between power failures (intentional and normal shut downs) by a 9 VDC supply which is plugged into your AC outlet. This is why the unit is reliable until the AC power to the house drops.

Since the MYARC unit has the basic 32K expansion memory already a part of itself, the 32K card in the PEB, or a similar 32K in the console, in a side car type unit, standalone, or in the speech unit, will have to go.

The MYARC unit will function with MYARC'S XBII to allow basic programs up to 128K in length. This could be a major advantage to someone who wants to write a very large program in basic and up to now been frustrated in the attempt. Remember also that you will have tremendous enhancements at your finger tips with that XBII, which does require the 128K memory, and with this advantage also comes the warning that there are still some bugs in that XBII program, and that if you do decide to use this package, the effort should be for yourself, since there is not an abundance of users with this same set up.

The SPOOLING feature is really a big plus for this RAM DISK. The unit will always have at least 80K of the 512K set aside for the spooler. Scott tells me that this leaves 400K for a RAMDISK (512K - 32K - 80K = 400K). It will decrease in availability, as the size of the spooler increases.

The print spooler is used by replacing the usual device names "PIO" or "RS232" with "SPPIO" and "SPRS232". No connection exists between the two cards involved in the printing process, the RAM DISK and the RS232 card. The MYARC card spools the data to be printed when instructed to do so by either command above, and sends it to the RS232/PIO card on an interrupt driven basis. This means that disk accesses will slow down this spooling process. Note that if your printer is equipped with a small printer buffer internally, you will never notice a pause. In any case the machine will function essentially as if it were used in conjunction with a large print spooler. The CORCOMP RS232 unit is not compatible with the MYARC RAMDISK.

The RAMDISK can supersede any other drive by executing CALL ENDK(n) where "n" is the drive number. CALL ENDK(0) will disable disk emulation. The RAMDISK can always be accessed through device name "RD".

Other calls include CALL RDIR, to list the RAMDISK

directory, CALL PART(400,80), or CALL PART(0,400) to partition memory between the RAM DISK and the print spooler, where the numbers represent thousands (K) bytes. CALL VOL("name") to rename the volume as well as other useful calls.

I would like to be able to tell you all the software that will run on this card as well as any other hardware it will not cooperate with, but in trying to keep these articles to one page and cover the essentials, I've decided to stop here before I have the opportunity to give you too much incorrect info.

The MYARC RAM DISK has been very well received by the T.I. community, and you can't go wrong in getting one. The intent in these articles is to give you an overall view of the various units, so you may be a little better informed when you do buy.

Next month I'll try to cover the CORCOMP "MEMORY PLUS". That is if one of the people using one can give me some feed-back on the unit.

Until next time, keep the Time.

Talking not spelling
by Jack Miller, Trenton Michigan
reprinted from MICROpendium, April 1987

In working with the TI speech synthesizer I came across an item I have not seen published in any of TI's manuals. This specifically refers to Extended Basic's use of CALL SAY or CALL SPGET.

If you refer to the list of resident vocabulary words in the Editor/Assembler or Speech Synthesizer manuals, you will see some phrases such as "Texas Instruments." A normal CALL SAY("TEXAS INSTRUMENTS") will cause the speech synthesizer to spell each word as though the words were not in its vocabulary. The secret to saying those listed phrases is to tell the synthesizer that it is to look for the whole phrase, not each word. This is done by using the # sign inside of the quotes before and after those phrase, i.e. CALL SAY("#.7#").

Along the same line, to get + to say "positive," - to say "negative" and . to say "point," you must precede these symbols with a # also unless the + or - precedes a number or the period is between two or more numbers. For example, to say "point seven" you must use CLL SAY("#.7#").

Another item that you may not have figured out is that almost all of the symbols when used alone say "uh-oh" during a CALL SAY. The exceptions are plus, minus, period, comma, colon and semicolon. These give no response when used alone. Preceding and following the above signs with a # sign will give the previously mentioned response for plus ("positive"), minus ("minus") and period ("point"), but the rest will then give "uh-oh."

TI-WRITER TRICKS WITH CLASS!!

by Edward Stamm BLUE GRASS 99 COMPUTER SOCIETY
reprinted from the HUGGER newsletter.

Some of you may use TI-Writer enough to be extremely proficient at it, while others may know just enough to get by. By necessity, I've had to be counted among the former. With my trusty Gemini 10X, I've written dozens of papers as well as a Master's Thesis with TI-Writer. Along the way I've picked up a few nifty tricks that make TI-Writer perform like the software written for Big Blue. I thought that I might share a few of these tricks with you. All of you will know some of what follows, and some of you may even know everything, and still others may prefer to do things differently. But some of these tips may benefit enough readers to make this article worthwhile. Many of the hints which follow will work only with Gemini and compatible printers, so be forewarned.

IN THE BEGINNING

When writing a paper, I never worry about formatting commands--- at first. I set the tabs at 2 and 38, so that all text can be viewed within the monitor screen. I then proceed to type to my heart's content, knowing that the formatting commands will be added later. When text is typed this way, you will run out of memory in about 630 lines or so. Don't type that much. Stop around 400 lines, save it, and start a new file. This is convenient for two reasons. First, it takes forever to save and load a 600 line file about 30 seconds per 100 lines! While it really doesn't matter (technically) if a file is broken up, this may prevent comprehension difficulties when you are composing the text.

"MASS" files

After typing the text, and saving it (calling it say "DSK1.XXX"), I call up my "MASS" file (I literally call it my "DSK1.MASS" file for lack of a better name), which I always keep nearby. As many of you know, one file (when printed) can call up another using the "Include File" (.IF DSK1.FILENAME) formatting command. To ease things a bit, I put nearly all my formatting commands in one file, with the last line being ".IF DSK1.XXX". The formatting commands I use (but you may prefer others) are:

```
LH 9, .RH 69, .PL 58, .FI  
LS 2, .HE, .FD, .IN 5  
.IF DSK1.XXX .IF DSK1.YYY
```

This will give double-spaced text, about 25 lines long, nifty margins, with page numbers, no less. Don't forget to load the file "DSK1.MASS", not "DSK1.XXX" when formatting.

Embedded Commands

You are easily able to embed two printer commands, besides the familiar & (underscoring) and @ (overstriking) commands (others can be embedded, but with more difficulty; these will be discussed later). These two commands are emphasized and double-strike. The "Emphasized" command is "ESC E". To embed "ESC E", type the following: "CTRL U", "FCTN R", "CTRL U" (these three keystroke combinations will henceforth be called "ESC") "E". You will see on the screen a funny character and an "E". Don't put a space bar between these characters. These will not be printed, and unlike "&"

and "@", they will work even if you print the file (PF) while you are in the Edit Mode. Everything following these symbols will be emphasized. To cancel the command type "ESC" "F".

IMPORTANT NOTE!

If you do not intend to "fill" your text (i.e. include the formatting command .FI), or if you are printing in the Edit Mode then watch out, Embedded commands exist in the eyes of the monitor but do not exist in the eyes of the printer. Consequently, all printed text will be moved two places to the left (at least for the line of text in which the embedded command appears). Embedded double-strike is "ESC" "G" and is cancelled by "ESC" "H". This command differs from "@" in that "ESC" "G" continues to double-strike until commanded to stop. The same warning concerning moving text applies here too.

TRANSLITERATE STATEMENTS

For some documents, I add a few "Transliterate" commands (.TL x:y,z). A transliterate statement enables the printer to print out y,z whenever it encounters x, without actually printing x. As an important example, think about "underlining". No, not underscoring, but underlining. For those of us with dot-matrix printers, this is a big difference. Underscoring merely puts a line under each letter, while underlining puts a continuous line under whole word(s). To do this, include these two formatting lines:

```
TL 60:27,45,1 and .TL 62:27,45,0
```

With this whenever "<" is encountered, the printer begins to underline, and will continue to do so until it encounters ">". Therefore, if these commands are used, <these words> will be underlined, while these words will be underscored.

ANOTHER IMPORTANT NOTE:

If your left margin is other than "0" then, you will have annoying underlines in the left margins. To avoid this, there are two remedies. One is to print out your document first, and then add ">" at the appropriate places to cancel the underline and "<" at the beginning of the next word to begin it again, and so on. The second remedy is to place your tabs at 9 and 69 when you are typing your document (using those annoying windows) and placing the "<" and ">" at the appropriate places in the text (and make sure that you do not "fill" your document). I find this to be the easiest method. Because of the hassles, underlining is most ideal for using with words or short phrases only.

Maybe you want to italicize certain words in your text. Put these .TL commands in:

```
TL 123:27,52 and .TL 125:27,53
```

With these commands, everything following (will be italicized and everything following) won't be. This is great if you want to italicize just one word in the text for stress.

Perhaps you need subscripts for footnotes:

```
TL 94:27,83,0 and .TL 92:27,84
```

When "*" is encountered, every thing following it is in subscript. The command is stopped when "&" is encountered. If your text is not being double-struck, then after "&", you should add "ESC" "H", since the superscript command automatically begins double-strike. To resume emphasized

Continued on page 5

Continued from page 4

print, you must add "ESC" "E", since superscript automatically cancels it.

STUPID PET TRICKS

I have found a few unusual tricks that are useful on occasion.

a. CTRL Y. This will enable you to backspace beyond the left margin as you are typing. Mention of this ability is buried in the TI-Writer manual.

B. CTRL K. This deletes, in one stroke, everything to the right of the cursor.

c. CTRL V. This moves the cursor, in one stroke, to the beginning of the line.

d. BEL. When I have a long document (30 + pages), I sometimes transliterate some symbol I don't use (e.g. "!") into ASCII "7" (BELL or BEL). I then put ! at various places in the text. Then, when the text is printing, and I'm in the kitchen cooking hotdogs, I know exactly where in the text the printer is printing. "!!!!" will make the printer ring four times.

e. After you have proofread a hardcopy of your text, and want to make a change in your file, it is often difficult to find the exact place in your file where to make the change. Use the Find String Command, "Seach" for an unusual or capitalized word in the vicinity of your error (or search for your error if it is a misspelling). This is easier than reading everything on the screen.

f. Using the transliterating style demonstrated earlier, one can change the printing pitch at will anywhere in the text.

(from the Bytemonger, by Edward Stamm.)

Steps to guard against mail fraud
reprinted from MICROpendium May 1988

While most companies doing business by mail are reliable and reputable, mistakes can happen in someone's order. Or, there can be outright fraud.

Postmaster Harold Creed of the Round Rock, Texas, Post Office outlined for MICROpendium some steps a consumer can take in order to prevent being victimized.

* When ordering anything by mail, Creed says, keep a copy of the order, showing the date and the address.

* If your order does not arrive within 30 days, write an inquiry letting your seller know that the check has cleared the bank and you have not received the merchandise. Again, Creed says, keep a copy with the date. Give the vendor 10 days to reply.

* If the order is not filled or if there is no explanation, bring in the copies of your records of the order to your local postmaster. "records are the big thing" Creed notes. The postmaster will normally make a set of copies to send to the postal inspector in the geographical area you ordered from.

* The inspector will normally tell you to write the vendor again, informing him that you have turned over the matter to postal authorities. It "doesn't hurt" to say you are going to turn the matter over to them in your first inquiry, Creed notes.

Creed says that the postal authorities eventually investigate all complaints, but "it takes so long to build a case on them it's kind of frustrating". If there are a lot of inquiries about a particular company, the investigation will receive a higher priority, he says.

A good way to check a company before you send money, he says, would be to inquire with the Better Business Bureau. He says it is a good idea to make a complaint to the BBB at the same time you turn records over to the postal authorities.

If the company is outside the United States, the procedure is "pretty much the same, but it might take longer on your answer," he notes.

If the company does not have the product in hand but cashes your check to use in making it, this is a "gray area-I'm not a lawyer," Creed says. The wording of the advertisement might be revelant, he notes.

The postmaster may not know the answer when a consumer has a mail-order problem, but will work with you, he says.

How to save program from tape to disk
by Jerry Keisler
reprinted from MICROpendium Dec 86

Re: "program Problems" (Feedback, Nov 86) , I also had problems saving long tape programs to disk.

The way the system acts I assume the first byte of data saved to tape tells the computer the length of the program. When the program is loaded, if this length exceeds the available console memory the load attempt is aborted.

To save long tape programs to disk I turn off the system, turn off the disk drives, then turn the console back on. If the program will load into Extended BASIC I use it, otherwise I use BASIC (Extended BASIC) works faster when deleting lines). Start at the top of the program list and delete about half the lines and resave the program to another tape. Now load the original tape program, remove the last half of the program listing and save it to a second tape. Turn your system off. Turn the disk drives on and turn the system on.

Load the first half of your tape program and SAVE
DSK1.A,MERGE

Load the second half of the tape program, then MERGE
DSK1.A

Now save the complete program to disk.

If you have a GRAM Kracker with GRAM Utility 1 loaded you can use the DEL starting line-ending line to save a lot of time.

FILE PROCESSING
reprinted from SNUGLETter

File processing on the TI is not as difficult as you might believe. The hardest part for me was figuring out the "examples" that were in THE owner's manual. They all went something like this:

```
100 OPEN #2:"CS1", INTERNAL, INPUT, FIXED
-
-
program lines
-
-
290 CLOSE #2
300 END
```

This in my opinion falls under the heading of "poor documentation". What was left out was the most important part! I tried and tried to get my computer to process files. I failed because I didn't know what to tell the computer to do with the files once it was open. I couldn't get past the mental block that told me "file processing is different from programming". In fact, programming is just a form of file processing.

The TI 99/4A handles ALL input and output through files. Most of the time, we are completely unaware that we are dealing with a "file" while programming. Page II-119 of the User's Reference Guide states "All TI BASIC statements which refer to files do so by means of a file number between 0 and 255 inclusive"..."file 0 refers to the keyboard and screen of your computer and is always accessible..." Since file 0 is always accessible. Statements such as PRINT, INPUT, RESTORE, etc which refer to the keyboard or screen do require a file number with them. You can however, write a statement such as:

```
100 PRINT #0: "print this to screen"
```

and have it do exactly the same thing as:

```
100 PRINT "print this to screen"
```

You can also INPUT from file #0, but since file 0 is always open, statements like OPEN #0 or CLOSE #0 will generate an error message.

All other open files must be referred to by their number. Remember that this number is only used by the program to remember which file is which and is not a part of the file at all. As a matter of fact, you could open a file with one number, process it somehow, close it, and then reopen the same file with a different number... all this within the same program!

Now that I've got you thoroughly confused, I'll give you a short sample file processing program to try to clarify what I've been saying. Most of us think of a file as being a disk or cassette. While these are indeed files to the

computer, they are by no means the only ones we have available. This short program opens a file to the Speech Synthesizer, sets up a FOR-NEXT loop to print a couple of sentences to both the screen and the Synthesizer, and then closes the file. You will need a TE 2 module to run the program. If you don't have a TE 2 just change the file name in line 110 from SPEECH to PID or whatever your printer requires. This will give output to the screen and the printer instead.

```
100 CALL CLEAR
110 OPEN #1:"SPEECH",OUTPUT
120 FOR Y=1 TO 7
130 READ I$
140 FOR X=0 TO 1
150 PRINT #1:I$
160 NEXT X
170 NEXT Y
180 CLOSE #1
190 DATA THIS IS A TEST OF THE SCREEN AND SPEECH FILES
ON THE
200 DATA TEXAS INSTRUMENTS 99/4A HOME COMPUTER. IT
SHOULD HELP
210 DATA TO DEMONSTRATE HOW ALL INPUT AND OUTPUT IS
TREATED AS A FILE BY THE COMPUTER
```

In this program line 110 OPENS a file to the speech synthesizer (or printer). Lines 120 to 140 set up some loops to read from the DATA statements and switch between files (0 and 1). Line 150 PRINTs the output to both outputs (0 and 1). Lines 160 and 170 increment the loops. Line 180 CLOSEs the computer's association with file #1, and lines 190 to 210 are the DATA read by line 130

The point is that the lines between 110 and 180 are the ones that do all the work. Whether you are working with a file or just printing to the screen, the programming is the same. All you have to do is tell the computer where you want the data to go to or to come from.

Try modifying line 110 from OPEN #1:"SPEECH",OUTPUT to OPEN #1:"DSK1.TESTFILE",OUTPUT. This will cause the second output (remember that #0 is going to the screen) to go to a disk in drive #1 under the filename of "TESTFILE". Try some other experiments in line 110 like using "CS1"; "PID" or "RS232" instead of "SPEECH". These will cause the output to go to the cassette recorder, printer, or modem respectively in addition to the screen.

Once you have mastered OUTPUTing to peripheral devices, the next logical step is to learn how to get INPUT from them. Some devices, such as the printer or speech synthesizer, by their very nature are one way devices. Trying to get input from them would surely lead to hours of frustration. Keeping that in mind, we will concentrate on the devices that have two way communication with the computer. The disk drive and cassette recorder are the primary devices we use for file storage. My experience with cassette based files has left me somewhat dissatisfied.

Continued on page 7

Continued from page 6

While there are provisions for storing SEQUENTIAL files on cassette, it is a cumbersome operation at best. There also seems to be a bug in the I/O routines for input from cassette. If you do any file storage and retrieval from cassette, keep in mind that the delay between the prompt:

```
PRESS CASSETTE PLAY (S) THEN PRESS ENTER
```

and the actual reading of data is longer in most cases than the tone leading to the data. I have found that if I press ENTER first then wait for the screen to scroll up 1 line before pressing cassette play that I have no problems. If you don't do this the computer may miss the beginning of the file and give an error.

Since getting input from cassette and disk is very similar I won't spend any more time on cassette.

Getting input from a disk file is almost the same as sending output to it. First you have to OPEN the file to the disk. This is done exactly the same as before, except instead of "OUTPUT" following the file name, we use "INPUT". The words INPUT and OUTPUT are two of the four modes that can be used to open a file. The third, UPDATE, is the default and means you can either read from it or write to it. If you don't specify one of the 4 modes, UPDATE will be assumed by the computer. The last mode is called APPEND and will only allow OUTPUT to the end of a file. Let's look at our program again. If you haven't already done so change line 110 to OPEN #1:"DSK1.TESTFILE",OUTPUT and run the program. Type in the new program below (or modify the old one to match).

```
100 CALL CLEAR
110 OPEN #1:"SPEECH",OUTPUT
115 OPEN #2:"DSK1.TESTFILE",INPUT
120 FOR Y=1 TO 7
130 INPUT #2:1$
140 FOR I=0 TO 1
150 PRINT #1:1$
160 NEXT I
170 NEXT Y
180 CLOSE #1
190 CLOSE #2
```

The main differences between this program and the first one are that we have added a second file number and name to the program (line 115), changed the "READ X\$" to INPUT #2:X\$, and deleted the data statements at the end of the program. We are now getting the data from the disk file that we just saved under the name of "TESTFILE", and #0 means the keyboard and screen. File #0 is an "UPDATE" file, #1 is an "OUTPUT" type file and #2 is an "INPUT" type file.

This has been very basic stuff so far, but in order to learn "FILE PROCESSING", you must understand the basics of how your TI-99/4A computer communicates with its peripherals. Once you figure out that the computer treats EVERYTHING as a file, you will be on your way to writing your own file processing software.

SIMPLE FIXES ON A DOT MATRIX PRINTER

Reprinted from the SUNCOAST BEEPER

Reprinted from the SUNCOAST BEEPER

PROBLEM: POOR PRINT PROBLEM.

SOLUTION: Check to see if the lever is adjusted at the wrong setting. Dot matrix printers have an adjustment system that allows the operator to adjust the position of the print head relative to the paper and platen to accommodate various thicknesses of paper. If this adjustment is incorrect, the printer may print too lightly. Another symptom of misadjustment of this lever is poorly formed characters. Most of the letters will be formed properly, but some dots probably at the top or bottom of the letters will be too light or not print at all.

Another possible cause of too light printing is perhaps more obvious. If adjustment of the adjusting lever does not solve the problem try replacing the printer's ribbon. If the problem seems to consist of a combination of printing too lightly and complete absence of printing of some portions of characters. Check the ribbon path. If the operator has inadvertently failed to thread it through all the proper points, it may not always be between the print head and the paper causing failure to print. If the printer has been used heavily for some time, the problem may be simply that the print head has worn out. In such a case if all other steps to restore proper print quality have failed, replace the print head. This is really a pretty simple operation. The operators manual for the Star 56-10 suggest this sequence of operations:

Turn off the switch, unplug the power cord and if necessary wait until the head cools off.

Remove cover and ribbon

Unscrew the two screws securing the print head.

Disconnect the print head cable.

Connect the print head cable to the new print head. Put the new head in position and fasten it with the screws removed from the old head.

Apply screw locking adhesive to the heads of the screws.

In many dot matrix printers there are two motors: one to drive the print head and one to drive the carriage. If the print head malfunctions, for example, if it fails to move, examine the belt that connects it to the drive motor during operation. If it is defective, replace it. Check the motor itself for proper operation. If the motor and belt are operating properly, look for evidence of binding or obstruction of the print head.

The other motor drives the platen and advances the paper. It is connected to the platen by a set of gears. If the platen fails to feed during printer operation, check for proper operation of its drive motor. Check for correct engagement of the gears.

SOUTH BAY
TI
USERS GROUP

SOUTH BAY TI USERS GROUP

P.O. Box 23447
San Jose, CA 95153-3447



DALLAS TI HOME COMP GROUP
P.O. BOX 29863
DALLAS TX 75229
Dec-99