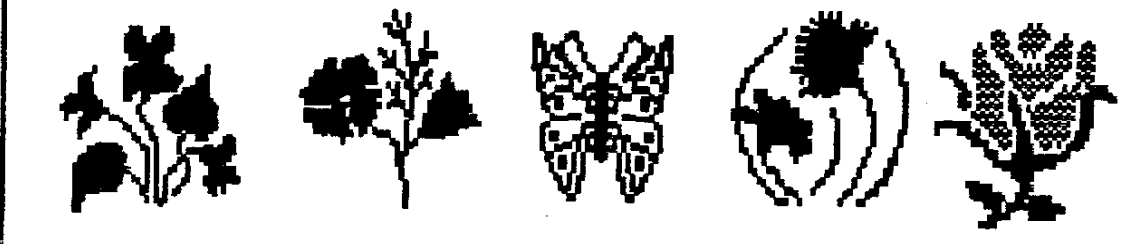
TI-D-BITS

PHILADELPHIA AREA USERS GROUP NEWSLETTER
COVERING THE TI99/4A
AND MYARC 9640 COMPUTERS

MAY 1988



THE PHILADELPHIA AREA TI-99/4A USERS' GROUP (MAY, '88)

The Philadelphia Area TI-99/4A Users' Group meets twice a month. On the first Saturday of any given month, we meet at the Bucks County Youth Development Center, (YDC, which is next to Neshaminy Mall), Administration Building, beginning at 10:00 am. On the third Saturday of each month, we meet at LaSalle University, 20th Olney, in room H-329 located in the Science Building. Membership to The Philadelphia Area TI-99/4A Users' Group is available to all. We invite anyone that is interested in the TI-99/4A to visit us. Stop in and see what is available to you for your TI and how membership can benefit you!

Current executive board consists of:

PRESIDENT.....	Don Arsenault.....	215-368-0446
VICE PRESIDENT.....	Allan Silversteen.	215-885-7910
SECRETARY.....	Mark Wannop.....	609-365-1776
TREASURER.....	Tom D'Annunzio....	215-947-7353

Committees consists of:

TI-d-BITS	Ralph Field.....	215-362-2534
	Don Arsenault.....	215-368-0446
	Bill Hughes	
	Rice Hall	
LIBRARY	Ted Cheney.....	215-752-1458
	Newton Stallman	
MEMBERSHIP ...	Bill Hughes	
ASSISTANT TREASURER.	Frank Passini	
EDUCATION	Barry Traver	
	Frank Passini	
	Ted Cheney	
	Tim Coyne	
	Carlo Angelico	
EQUIPMENT	Rice Hall	
PROGRAM	Dr. Eric Bray	

REMEMBER to be considerate when calling any of the above people. Limit your calls to the early evening hours. (6pm to 9pm)

The opinions expressed herein are those of the individual authors are not necessarily those of the Philadelphia Area TI-99/4A Users' Group or its officers. Nor is the Philadelphia Area TI-99/4A Users' Group or any of its officers responsible for any damage, inconvenience, or loss which may result as a consequence of the use of any written material herein.

TI-d-Bits is published monthly by the Philadelphia Area TI-99/4A Users' Group, P.O. Box 18486, Philadelphia, PA 19120. All material herein may be reprinted freely by other non-profit User Groups, (unless otherwise stated), as long as proper credit is given to both source and author. Contributions are encouraged, but no payment is made. Editorial, advertising, and classified, copy MUST be in by the LAST day of the previous month. You can either mail your copy to: TI-d-Bits, c/o The Philadelphia Area TI-99/4A Users' Group, P.O.Box 18486, Philadelphia, PA 19120 or send it via modem by contacting Don Arsenault at (215)-368-0446. If your piece contains any diagrams, charts, or code, send a paper copy AT FINAL PUBLICATION SIZE.

Classified ads are printed in blocks. A block consists of 3 lines, 55 characters wide, or any increment of 3 lines. Classified advertising is accepted from members at NO CHARGE for a one block ad, per issue. Additional ads from members may be placed at cost of $1.00 per block. Non-members may place classified ads at a cost of $2.00 per block. All advertisements MUST be paid for in advance.

Commercial advertising is accepted for publication at the following rates:

Quarter page	$ 5.00
Half page	$ 8.00
Full page	$15.00

Commercial advertisements will be placed in the next available issue. All advertisements MUST be paid for in advance.

The editor of TI-d-Bits or the executive board of The Philadelphia area TI-99/4a Users' Group reserve the right to reject any material submitted for publication for any reasons.

The Philadelphia Area TI-99/4A Users' Group's program library is available to all active members at NO CHARGE for copying to your disk. A charge of $2.00 per disk is made for club supplied disks for members. Non members may obtain copies of the library for a fee of $5.00 per disk. A catalog of the library's contents is given to all new members upon request and updates will appear in this publication from time to time. To obtain material from the library, contact the librarian for the best procedure to obtain your requests.

PRESIDENT'S COLUMN

There has a lot of activity this month in regards to the club, which has taken up much time, so this column will be short and sweet.

First I would like to thank Frank Passini for his donation of a "BIG BLUE" printer to the club for use with our system(s). The printer is a thermal transfer printer which was originally sold for use with the IBM PC. It should work well with our equipment, in fact it is already hooked up to the system. All we need *now* is a source of thermal printer paper for it. If you know of any suppliers please let us know.

Ron McKeever has volunteered to conduct a series of study sessions on the use of The Printer's Apprentice. Ron has been using this package in conjunction with one of his college courses, and is well versed in its use. Now is the time for you to get your copy of this fine publishing program and attend the sessions. You do not need to have the program for the sessions, as we have enough personal copies of the program for use at these sessions. The sessions will be held at the YDC meetings starting with the June meeting. Maybe we can get enough people interested and knowledgeable with this program to get some help with the newsletter (HINT..HINT..).

We have just acquired for the club a substantial amount of new software, amounting to about 450 disks full. Some of it is commercial and some is public domain and freeware. The freeware and public domain stuff will be put in the library, and the commercial programs will be offered for sale at greatly reduced prices. More info on this a little later.

X$)X$)X$)X$)X$)X$)X$)X$)X$)X$)X

SECRETARY'S NOTES
By W. Mark Wannop

SO WHAT HAPPENED LAST MONTH?

To anyone who missed this rambling last month, my apologies... I caught some kind of nasty bug and was "down" off and on for about three weeks (not my computer - me!) and I missed the Dreaded Deadline Doom... So I'll just cram most of it in here...

AS USUAL...

The first part of this column duplicates the DATA BUS column; those who've waded through it once already can jump down to "Notes From Here And There".

PACS COMPUTER FESTIVAL NOTES

Well, folks, I had the strangest feeling when I entered the building where the PACS Computer Fest was held... As I arrived about lunchtime, I decided to visit the cafeteria before venturing upstairs. I glanced about the lunchroom, and instead of the rabid computerists I expected to see, there were several dozens of Girl Scouts. "What?", thought I, "Is this the PACS Computer Fest and Girl Scout Convention?"

Upon climbing to the ballroom, the world of PACS returned to what passes as normal; many folks milling about, looking at the various computers that other chaps were cheerfully running their niftiest software on. And nary a Girl Scout in sight... Normalcy, as Mr. Harding would say...

I made my way to the tables occupied by the TI SIG where several TI-99/4A systems were up and running, along with the GENEVE 9640. Various programs were being demonstrated on both machines, and the group's public domain/shareware library was on hand. Many members were glad to take advantage of the library. Library disks were (and are) recorded for free if the TI SIG member supplies the blank disk; it's two bucks for members if the SIG provides the disk, and five bucks for non-SIG members - so join!

(The newest disks from the library are available at our regular meetings for member's convenience, and the entire library is available for copying at the YDC meetings on the first Saturday of each month.)

We got some very nice comments from various PACS members on our machines, as well as some surprised ones; some folks were genuinely surprised to find that there was "still" an active group of TI users, were surprised by the quality of the current software, and were surprised by the GENEVE! (These folks must get about two blank pages a month in their DATA BUS, where this column normally sits...) We also got some fairly sad comments, as several folks mentioned that they once owned a TI system and then sold it when TI left the "home computer" marketplace, and that they regretted getting rid of it. I can personally understand this feeling, even though I retained my TI... I own two "other" systems and have access to a third; I find all these computers interesting and enjoyable to work with, but if I could only have one system, I would stay with the TI. The 99/4A has its quirks like any other system, but I've found it the

easiest to get along with of all I've used. Anyway, I like to keep my old friends!

And speaking of other systems (and friends), I hope everyone had a look around the ballroom at what the other SIGS were doing. There were many impressive displays going on all the machines present. It was interesting to see the various ways people use their machines, and in some cases, how they've altered their machines to suit their purposes. Also, I was glad to meet in person some of the folks I've been cooresponding with on the BBS's, such as Bob Toscani of the COCO SIG.

T.I.C.O.F.F. - A SORT-OF TI FEST

Exactly one week after the PACS Computer Fest came the annual TICOFF at the Roselle Park High School. This year, in an effort to broaden the base of the show (proceeds go to a scholarship fund) it was opened to other computers as well, although the speakers were mostly, if not all, from the TI community. PATIUG founder (and excellent programer) Barry Traver gave a talk, as well as J. Peter Hoddie (another excellent programer) and Lou Phillips (head of Myarc, manufacturer of the Geneve), and other fine folks.

I was a tad disappointed by the sales area on two fronts; first, the amount of TI vendors was down sharply from last year. It isn't that the missing vendors are no longer in business; I believe some were discouraged when TICOFF opened up to IBM clones, and were afraid it would wind up like every other computer show. On the other hand, the vendors with the non-TI materials had a fairly slow day, as there weren't all that many non-TI users in attendance. In fact, some of the non-TI customers were confused: "What's with all this TI stuff all over the place?" one chap grouchily asked me. You should have seen his face when I told him TICOFF stood for "TI Computer Owners Fun Fest"...

APRIL MEETING NOTES

Now onto the new stuff! Several important announcements were made starting off the meeting: NO poster designs were submitted... We need a nice design to be hung in computer stores, supermarkets, etc. advertising our club; let's all (including me) get off the stick and design a poster or two! It came to the attention of the officers that the printer we have been using at meetings does not belong to the club but belongs to Rice Hall, who has been kindly bringing it along as he brings the club equipment! (Rice is our Equipment Chairman, which means he gets the honor of lugging that heavy stuff around.) We would like to request a donation of an unwanted printer from a SIG member (or possibly a donation to a "printer fund"...) In the same vein, a printer ribbon re-inker purchase was discussed, without much interest... All in all, the group was very happy with the turnout at the PACS computerfest! And Steve Weissman, SysOp of the PACS BBS, will be coming to a future meeting to discuss the BBS, how it functions, and What You Should Know to use it! With the way modem prices are dropping, more of us could get into telecommunications!

PLATO EDUCATIONAL SOFTWARE ON THE GENEVE 9640

The Myarc Geneve will run most TI software modules that have been "dumped" to disk using either the program supplied with the Geneve or with a Gram-Kracker; but a few modules do not run correctly. Myarc has been steadily providing "patches" to correct operations on the most desired modules.

Dr. Eric Bray demonstrated the PLATO module, which allows the user to load PLATO educational programs from disk (PLATO is copyright Control Data corporation, and available for a wide range of computers). This was one of the "problem" modules, and the patch supplied by Myarc worked flawlessly. Eric displayed programs on the human body, history, and French. There is a range of roughly 100 different programs on disk available for the TI, ranging from grade-school level through high school. This module and its programs are a valuable learning tool for students and for us older folk who want to brush up a tad as well; it's good that Myarc is providing support for older programs as well as providing some great new ones!

TELCO COMMUNICATIONS SOFTWARE

Don Arsenault showed off some new programs from our library; I'd like to briefly rave on about TELCO! It is, in my opinion, one of the nicest communications packages I've seen for the TI. Unlike many TI communications programs, this one runs well on the GENEVE 9640, and has features included specifically for that computer. This is a shareware offering from a Canadian programmer; suggested donation is $20.00, and the program is worth much more than that.

The first striking thing about the program is its appearance; while released as "shareware" the programs appearance is every bit as professional as any commercially release software package. Selection from the various menus can be made with either the arrow keys/enter combination, or by pressing the number or first letter (as the case may be) of the option desired. Menu selections are highlighted by inverse bar text; while this is common on many other systems, I don't recall seeing this technique used on the TI before.

The program is easily set up by the user to conform to his needs. There are three terminal emulations to choose from, baud rate and parity are easily set, as well as

foreground and background colors. An information bar appears on the bottom of the screen, giving elapsed time (reset at each log-on), baud rate, parity, terminal mode or program function, and state of the text buffer.

The text buffer can be set to save to disk when it fills, and to automatically increment the filenames of the buffer saves. File transfers available are Xmodem up/down loads and ASCII upload (the text buffer is the "ASCII download". The file transfers work well. One nice feature is that pressing FCTN 7 ("AID") gets you a help screen pertaining to where you are in the program.

I only have two misgivings about the program; one is that the printer spooler only supports PIO, and I'd like to spool to my 80 column terminal, which is RS232 (this, obviously, won't be a concern to most other folks....). The other problem is the documentation; the docs are well written (and there are plenty of them) but some invisible page feed characters have gotten into them, causing the printer to spew paper crazily. I managed to correct the problem using Funlwriter, but it took an evening. I've heard that later docs are available; at any rate, I'll make my "debugged" version available as well.

TERRWARE AND "WHEEL OF FORTUNE"

The next new library disk shown was TERRWARE, also a fairware disk. This one consists of three programs. JOKER POKER is a nice poker adaptation with good card graphics. Game play is easy and fast. BLACKJACK is, of course, blackjack; this too is nicely done with attractive graphics, and features joystick control of bidding, hit, stand, double, and so on.

The above two programs would make a nice addition to the games section of anyone's library, but there is a VERY nice adaptation of "WHEEL OF FORTUNE" in the package! With excellent graphics, sound, and speech, the program simulates game play of the TV show very closely, selecting phrases from a variety of catagories. Up to three people can play, and the one with the most money at the end of three rounds does the timed "bonus round." No, you don't get the car (and Vanna isn't there either), but it's a heck of a lot of fun!

WORDS AND MUSIC BY RODGERS AND HAMMERSTEIN...

When I heard TI-SING6S, I thought it was a howl; but when I heard SOUTH PACIFIC, I thought that was a DOUBLE howl! This disk includes three songs from the musical both played and SUNG on the TI computer (speech synthesizer necessary)! The programmer is Ken Gilliland, who must have stayed up some very late nights to put this one together... For each of the three songs, a different graphic is displayed, the music is programmed in three part harmony, the speech synthesizer SINGS the lyrics ON KEY, and the character shown in the graphic opens and closes his mouth precisely timed to EACH SYLLABLE! How well does it sing? Well, it won't replace Enzio Pinza... Heck, it sounds FUNNY!!!

Included in this package are BALI H'AI, SOME ENCHANTED EVENING, and YOUNGER THAN SPRINGTIME; also there's a "read-me" text file which gives some backround on how all this works (worth reading!) PLUS there are two TI-ARTIST files; one that creates a "mini-poster" and one that creates "South Pacific" labels for your disk and envelope!

This is also a shareware effort; Gilliland plans LOTS more of these, including: SOUTH PACIFIC II, THE MUSIC MAN, WIZARD OF OZ (I wanna hear my computer sing "If I Only Had A Brain"!), OKLAHOMA, PORGY AND BESS, CAROUSELL, THE PATSY CLINE SONGBOOK I and II (!), and, believe it or not, WAGNER LYRICS FOR THE TI TENOR!

CLOSING ANNOUNCEMENTS

Ted Cheney announced that he will be holding classes in Microsoft Multiplan at the YDC meetings, at 9:45 a.m.; Don mentioned that someone is being sought to do a Printer's Apprentice seminar, as this is a very good, but rather complicated program.

Barry Traver announced that he has made a deal with ASGARD SOFTWARE (a "friendly competitor" of his own GENIAL COMPUTERWARE) to directly represent them at our meetings, and offering a 10% discount! Of course, Barry still represents GENIAL as well...

Eric Bray remains our link with Myarc; all Myarc peripheral cards for the TI or 9640 (including the 9640) can be ordered through Eric at the best price you're likely to find!

And coming up in May, it was announced that we'd have a visit from the COMPUTERS IN EDUCATION group!

NOTES FROM HERE AND THERE

TRENTON TOPICS

The Trenton Computer Show was held the weekend after the PACS meeting; Trenton is the largest show of its kind in the northeast. Unfortunately, the sporadic rain on Saturday made the flea market shopping erratic, and was followed by high winds on Sunday. Also, attendance was down sharply on Sunday, probably due to Saturday's rain.

There was supposed to be a TI meeting on Sunday, but from the reports I was given, it didn't happen... A former member of PATIUG showed up to sell his TI system, and

that was about it...

Regardless of that, there were buys to be had; 300/1200 baud manual direct connect modems were going for $20.00; 300 baud acoustics for a buck! The best deal I found on modems was the Apple 300 baud - despite the name "Apple" on it, it is a standard 300 baud Hayes compatible smart modem, and they were selling for $2.00. One fellow I know bought a dumb terminal for a buck; it looked in good condition, and even if it needed some repair, it was still a bargain!

I picked up some old COMPUTE!s to round out my collection, and found this bit of TI history; the earliest regular coverage of TI in COMPUTE! began in January 1983 with C. Regina's column, and in the March 1983 issue there were 5 TI articles published. The only mention I find of TI earlier is in the December 1982 issue, where various sort routines are discussed for the TI, Apple, Vic 20, and COCO...

HAMMING IT UP IN DAYTON

The weekend following Trenton (my god, am I writing a diary?) I found myself at the Dayton Ohio Hamfest. This show is roughly the size of the Trenton fest, and is about half computer oriented, the rest aimed at amateur radio. And at this fest I came across a TI dealer that I've never seen advertised before!

L.L. Conner Enterprise (1521 Ferry Street, Lafayette, Indiana 47904) has an eight page flyer of TI and TI related hardware and software. He seems to have a wide range of items. Some prices I thought very reasonable, others a tad high (this is, of course subjective - you should judge for yourself). You can order by phone or mail, he accepts Visa, M/C, and money orders, and personal checks if you don't mind waiting for them to clear...

There seems to be almost everything listed here one could want, even down to TI chips. While you can get many of them from local vendors (like the folks I work for), some are a tad hard to find; for example the 9918A VDP chip is a hard animal to track down - his price is $5.00.

I don't know if there is a charge for the flyer by mail, but a stamped self addressed envelope should be sent in any case. The phone numbers are (317) 742-8146 and (317) 423-4879.

DEPTFORD DOINGS

The Deptford TI group put out the first copy of its newsletter, the "99+ Express". This, by the way, is also the name of the group. In the newsletter is a column by yours truely called "Notes From Here And There"... Now where have I heard that before? Yep, a slightly re-written version of the "second half" of this one; instead of "Deptford Doin's" there's a section called "Philly Features"! I figure this way we can aid communication a tad. If anyone has any notes or tips they'd like to share with both groups, let me know..

Tony DiFebbo announced at the last meeting that he will be moving to Florida in the end of May; he will be missed by folks here in PATIUG as well as Deptford. John Simpkins will take over the President's chair, and a new Vice President will be selected.

ATLANTIC CITY ANTICS

I managed to visit the Jersey Shore chapter of DVUG just before sending this column over the phone lines... They have set up a new BBS. Called the Shore Line BBS, it is currently using TIBBS, but may change programs in the future. The BBS number is (609) 965-7930.

BITS FROM BRITAIN

No, I did NOT go to England... I mentioned in a previous ramble that there was a computer called the "Cortex" which was somehow TI related; here is a bit more on it, quoted from "International TI Lines":

"The Cortex is a 9900-based micro, a project from Electronics Today International, and its Power BASIC is very similar to TI's."

I also mentioned a replacement EPROM for the TI RS232 card being produced in England in a previous column; the same newsletter points out that this EPROM may make it possible to use serial functions with the speed-up crystal modification to the TI, as it can be set to ANY baud rate - just have to find out which one works!

And Now For Something Completely Different...

THE CRYPTOLIMERICK!

T WCTDWUQCWW GS LTDC IA DTVQQHCQ

HCWNVLCH WQWUW AIQ URX DCL GS FXDIQUCQ

GQU WYC FXQMHL'U IQXVQTD

T UYWLV RIQUY T HTDL

UX WQWU YCQ URX WQWUXQW UX WQWU YCQ

Hang In There...

)(*)(*)(*)(*)(*)(*)(*)(*)(*)(*)(

HOME COMPUTER USERS EXPERIENCE EYE PROBLEMS

Eye problems experienced by many people using computers on the job may be spreading into homes, says the Pennsylvania Optometric Association (POA).

"When used for lengthy periods of time, home computers can put as much stress on the eyes as the office types," said Dr. Scott Edmonds, President of the association.

Computer related eye strain symptoms include: headaches, blurred vision, itching/burning eyes, fatigue, and flickering sensations.

"Studies have traced the causes to vision problems computer users already have, and to improper working environments," Dr. Edmonds said.

"Sometimes you can have a minor vision problem that doesn't interfere with normal seeing tasks. But it will show up under stress of staring at a computer screen for hours at a time," he added.

Annual eye examinations are recommended for people who use computers at home or at work. Sometimes, glasses with a mild prescription are needed to reduce stress when using a computer. For those who already wear glasses, it may be necessary to have a specially designed pair for computer use.

"Be sure to talk to your optometrist about how much you use your home computer and any eyestrain symptoms you are having. Also, measure the working distance between youe eyes and the screen. All of this is important in determining your computer-related eye care needs," Dr. Edmonds explained.

In addition to seeking regular eye care, home computer users can help their eyes by taking 10-15 minute rest breaks every hour or two; eliminating glare from windows and lights; reducing room lighting; using adjustable furniture; and placing reference material next to the screen to avoid frequent eye and head movements and focusing changes.

For more information and a free pamphlet titled, "VDT User's Guide to Better Vision" write the Pennsylvania Optometric Association, P.O. Box 3312, Harrisburg, PA 17105.

X(X)X(X)X(X)X(X)X(X)X(X)X

PUTTING IT ALL TOGETHER #3
by Jim Peterson

The hardest part of learning to program is not in learning what the various commands do - it is in learning how to put them together to do what you want them to do! Key in this mini-program and run it to see what it does. Then read the explanation of each line and see how it does what it does.

```
100 !FORMATION by Jim Peter
on - use the S and D keys
110 CALL CLEAR :: CALL CHAR(
100,"381010FEFE3838101038 3F
EFE1010383 8"):: CALL SCREEN(
5):: CALL MAGNIFY(2):: RANDO
MIZE
120 V,W,P=0 :: FOR J=1 TO 7
:: CALL SPRITE(#J,100,7,1,25
0#RND+1,10,4):: FOR D=1 TO 1
00 :: NEXT D :: NEXT J :: CA
LL SPRITE(#11,101,16,160,128
)
130 CALL KEY(3,K,S):: W=W+1
:: IF W=150 THEN 170 ELSE IF
W=300 THEN 180 ELSE IF K=68
THEN V=V+2+(V)125)#2 ELSE IF
K=83 THEN V=V-2-(V(-125)#2
140 IF P=0 THEN CALL MOTION(
#11,0,V)ELSE IF P=1 THEN CAL
L MOTION(#11,0,V,#12,0,V)ELS
E CALL MOTION(#11,0,V,#12,0,
V,#13,0,V)
150 CALL COINC(ALL,A):: IF A
=0 THEN 130
160 CALL SOUND(1000,-4,0)::
Z=Z+1 :: DISPLAY AT(24,1):"P
LANES LOST";Z :: CALL DELSPR
ITE(ALL):: GOTO 120
170 P=1 :: CALL POSITION(#11
,R,C):: CALL SPRITE(#12,101,
16,160,C-40-(C(40)6):: GOTO
140
180 P=2 :: CALL POSITION(11,
R,C):: CALL SPRITE(#13,101,
16,160,C+40+(C)216)6):: GOTO
140
```

This is not a finished program but a mini-program to demonstrate the use of sprites.

Line 110 first clears the screen. The CALL CHAR lists a hex code string of 32 characters. The first 16 of these will reidentify ASCII 100 into an airplane pointing down, the remaining 16 will reidentify the next

ASCII, 101, to an airplane pointing upward. The screen is colored dark blue, the sprite magnification is set at 2 (single character in double size) and RANDOMIZE insures a different flight pattern each time.

In line 120, V and W and P are all set or reset to 0 (note that all can be included in one statement) because the program execution returns here from line 160 to restart after each crash. The J loop runs 7 times to put 7 srites on screen, numbered 1 to 7, using ASCII 100 (the down-pointing plane), colored red (color code 7), at dotrow 1 (top of screen) and at a randomly selected dotrow between 1 and 250 (thus each game will be different), moving at a speed of 10 downward and 4 to the right.

The D loop creates a delay so that each sprite will drift downward before the next is created, so that no two will overlap and be later detected as a coin- cidence in line 150; also, so that more than 4 will not appear in a row and be blanked out. After these have been placed, sprite #11 with ASCII 101 (the (16), is placed at dotrow 160, dotcolumn 128, without motion.

In the CALL KEY in line 130, the use of mode 3 insures that the ASCII of an upper case S or D will be returned even if the alpha lock happens to be up. W is a counter for the number of times that the keyboard is scanned (program execution passes through this line). When the count reaches 150, a jump to line 170 places a second plane on the screen, and at 300 a jump to line 180 adds the third plane.

Otherwise, if K=68 (D, the right arrow, is pressed), the speed of the CALL motion in line 140 is increased by 2. If this velocity is increased beyond 127 the program will crash, but if the relational expression (V>125) is true it will have a relational value of -1, and V=127+2+(-1)*2 will still be 127. If K=83 (S, the left arrow), the speed is decreased by 2, and the same formula insures that the velocity will not go below -127.

In line 140, P is the counter for the number of planes in the formation, initially set at 0 for one plane in line 120, increased to 1 and 2 in lines 170 and 180. This determines whether the CALL MOTION will change the speed (if any key was pressed) of 1, 2 or 3 sprites. The CALL MOTION has a 0 row velocity to keep the plane at the bottom of the screen, and a column velocity determined by the last keypress. Since all sprites are controlled in a single CALL MOTION, the speed change is simultaneous and they stay in formation.

In line 150, the CALL COINC checks whether any sprite is overlapping any other, even slightly. Since the red planes were all placed on the screen separate from each other and all traveling in the same direction and speed, they will never touch so any coincidence must be between a white plane and a red one. If there is no coincidence, A will equal 0 and program execution goes back to the CALL KEY.

If A=-1 there is a coincidence and line 160 creates a sound (I wish the TI was capable of a good bang!), the counter of crashes is incremented by one and displayed, all sprites are deleted, and execution goes back to line 120 to reset variables to 0 and place a new random formation on the screen.

When program execution jumps to line 170 from 130 after 150 key scans, the P counter for number of planes is changed to 1. CALL POSITION finds the dotrow R (not needed but must be included) and dotcolumn C currently occupied by the white plane. CALL SPRITE uses that value to place a second plane, sprite #12, 40 dotcolumns to the left.

If the original plane is less than 40 dotcolumns from the left of the screen, this would cause a crash because dotcolumn cannot be a negative number. The use of relational values again solves this problem. Suppose that sprite #11 is at dotcolumn 10. 10-40-(-1)6 will place sprite #12 at dotcolumn 226 which will be 40 dotcolumns to the left of #11 when wrapped around.

Similarly, after 300 keyscans, line 180 puts a third sprite-plane 40 dot-columns to the right of the first one.

This program could be modified in many ways. The number of red planes can be changed in the J loop in line 120 - if more are added, the D loop might also need adjustment to insure that none will wrap around before all are placed.

The rate of speed-up can be adjusted by changing the 2 in line 130 to some other value, even a decimal value such as 2.5. If you would rather use a joystick than the keyboard, change line 130 to:

```
130 CALL JOYST(1,X,Y):: K=X*
10+Y :: W=W+1 :: IF W=150 TH
EN 170 ELSE IF W=300 THEN 18
0 ELSE IF K=40 THEN V=V+2+(V
=V-2-(V<-125)*2
```

By using variable names rather than values for the J loop in line 120, or for the velocity in line 130, you could offer options of difficulty at the start of the program. When doing any programming with sprites in motion, it is always necessary to do a good deal of on-screen experimentation and program modification to get the desired results.

APPLE AND TI SIGN VAR DEAL

Austin, TX--TI is going to sell Apples, according to an agreement that's been inked between the two giant companies. Texas Instruments is now a value-added reseller of Apple, providing a workstation that closely integrated artifical intelligence technologies with mainstream personal computing.

The terms of the agreement call for TI's Data Systems Group to purchase Macintosh II personal computers from Apple, and incorporate a specially developed coprocessor board, The new board is based on TI's Explorer software environment. The TI group will market the new "microExplorer" systems with team selling and coop marketing support from Apple. It will be co-labeled with TI and Apple logos, something that both companies are calling a "marketing first."

The new computer to be marketed provides users with an AI microprocessor that's been optimized for development and delivery of symbolic processing applications such as expert systems, as well as access to the large base of conventional systems and personal productivity software that's already available for the Mac.

In commenting on the agreement TI president and ceo Jerry Junkins, said,"Combining Apple's strengths in personal computing and our own strengths in AI is a significent development that broadens the applications and business potential of AI technologies. We believe that many of our Fortune 1000 customers will find that this kind of cost-effective integration of symbolic and numeric computing will facilitate the widespread deployment of expert systems and other AI applications."

On the oher side of the fence, Apple's president and ceo John Scully said, "TI's Lisp coprocessor extends the Macintosh II into new applications areas that are complementary to our other Macintosh marketing thrusts. This is an important catalyst that should generate greater use of AI technologies in solving difficult business problems."

The TI Lisp chip is said to be the world's first microprocessor that's been designed specifically for AI. It also is used in the company's Explorer II computer, called the highest performance symbolic processing workstation. Use of the Lisp chip allows the microExplorer systems to provide higher symbolic processing performance than competing special purpose AI workstations, according to the two companies involved, and at significantly lower prices. List prices for the two systems start at $14,995 for the low-end entry-level configuration to a current cap of $26,795 for top-of-the-line models.

(8)(8)(8)(8)(8)(8)(8)(8)(8)(8)(8)(8)

GRAND RAM

Fm Charlotte TI Users Group

By now many TI users have either heard from their user groups or gotten information in the mail regarding the new "Grand Ram", a product from DataBioTics. Information provided by the Los Angeles based company together with files found on the General Electronics Information Network offer an interesting picture of the new product.

Grand Ram is a peripheral expansion box card for the TI 99/4A which is advertised as having "nearly limitless growth potential". First, it is a ramdisk. It is capable of emulating a single double-sided, double-density drive or two double-sided, single-density drives, or four single-sided, single-density drives, which enables you to use it for file management or to enhance disk copy operations. As a ramdisk, it has pluggable memory chips, which may be expanded as the buyer is able. And it is battery backed.

Second, the ram may be used as a print spooler--more commonly known as a print buffer.

Next there is also an optional real-time clock.

There is also an I/O (input/output) expansion port, which DataBioTics indicates will interface with future products. Technical specifications released August 14, 87 say that one expansion port is a cartridge emulator port, for support of a future add-on board which will emulate 99% of the cartridges in existence. The other port is a device expansion port, to implement multiple channel music boards, digitizing equipment, home control, and so on.

Should these devices not materialize, they are available from other sources, and DataBioTics includes with the technical specs the information needed for the appropriate hardware projects).

Promotional material for the Grand Ram indicates that it is accessible from any language. This was a problem

with some other products (not from DataBioTics) such as the Foundation 80 Column Card, which was advertised as being accessible from BASIC, and then wound up being useless for anything but the Z80A card.

The Grand Ram comes with source code, disk manager and terminal emulator software. The cost to build your own is as low as $100.00 ($99.95) for the 64K version, up to $185.98 for the 512K kit. If you order it assembled, the 512K is only $229.95.

More information on the Grand Ram is available from DataBioTics, at 1 800 255 2985.

(!)(!)(!)(!)(!)(!)(!)(!)(!)(!)(!)(!)(!)

==========

THE PRINTER'S APPRENTICE
(A basic beginners walkthrough)
By Rick Felzien
WEST JAX 99ERS

==========

When I first opened the documentation for the package called The Printer's Apprentice, I thought that I had acquired a book written in ancient Greek or some other strange language. The first strange thing that you notice is that menu selections are referred to as descendents. My first reaction was complete bewilderment. After reading through the book a number of times and referring back and forth where it said "See descendent", I finally began to make a little sense of the whole matter. After reading and re-reading and trying and re-trying the menu items, after a while things began to come together.

This article will hopefully help those of you who have purchased the package and are having trouble, as I did, in getting started.

The program set comes on four SS/SD disks. Disk one is a flippy with the programs on the front and some files on the flip side for use with the program set. Disks two and three are font disks, and disk four is Business Graphs 99 programs.

When making backup disks you must use a disk copy program as the program sectors must be the same as the original due to the programs being written in TI-FORTH. A double-sided version can be made if you use the procedures outlined in my article on making a double-sided copy of the PILOT-99 disk in a previous newsletter. The same goes for the Business Graphs 99 disk, you can copy the font files in the usual fashion.

When you first load the Printer's Apprentice, hereafter referred to as TPA, you get a six item menu, which are self explanatory:

1. Character Editor
2. Picture Editor
3. Formatter
4. Scheduler
5. Exit to Monitor
6. Exit to TI-FORTH

When you select Character Editor you have an editing screen with a menu across the bottom:

Edit; enters edit mode

Disk; allows you to get a (Dir)ectory of a disk, select a Filename, or eXit.

Convert; lets you convert Sdsh fonts (single-strike-single height) to Dush fonts (over-under--single-height). The unique program setup allows you to use over-under strike to get better quality fonts in the printer.

Setup; sets character style--Sdsh or Dush.

help; startup order; Setup, Disk, Print & variables then eXit.

Print; used to set up printer. The menu is as follows.

Printfile String Variables Go
Writeindex Readindex Extern Back

This menu is pretty self-explanatory. The following is a setup of the Variables menu and its defaults.

Printer	Eps	Gem	E
Density	Sd	Dd	S
Left margin			0
Right margin			480
Space (ASCII 32) size			4
Intercharacter size			2
Centerline (Y/N)			N

String allows you to print a string directly to the printer using the selected variables in the selected font to let you see how a particular setup will look.

NOTE: Readindex will read the width of each character in the font selected into memory to allow strings to be printed with the correct spacing for each letter and word.

NOTE: Writeindex will read the width of each created character or each character of a selected disk font into

an index and store it back to the font file--if this option is not used after editing or creating a font file, character spacing may not be correct.

By using separate filenames and changing Height and Width, you can change the size of an existing font and create a separate font file.

Extern creates a D/F 80 file for use with the Scheduler. D/F 80 files can be saved with TI-Writer by using PrintFile and when prompted for filename enter F DSKx.filename. By prefixing the filename with an F will create fixed file format.

When you select the Picture Editor you are presented a blank editing screen and must use the appropriate CONTROL/FUNCTION keys to obtain a specific menu (see the lists of keys included with this article). The picture editor is pretty well self-explanatory so I won't elaborate on that.

The Formatter is used to format the text and picture files to External format for use with the Scheduler. This allows you to set up such things as which font you want to use and the formatter even has its own text editor for creating text files rather than using TI-Writer and then converting to fixed format.

Upon loading the Formatter you get the following menu:

Go Dir Vars Hyphen Jotter eXit

Go; is used to set up the file for the Scheduler and is the last selection you will use in the Formatter.

Vars; lets you set up font type & style as well as other options that you will want to select for your formatted files. The following defaults are selectable.

Prntr Type / Eps Gem	E
Prnt Dnsy / Sd-Dd-Hs-Qd	S
FontStyle (Sdsh/Dush)	S
Linefeed size	0
Space (ASCII 32) size	4
Font/ASCII	F
Wrap/Fixed	W
Ragged/Microadjust	R
Left Margin	0
Right Margin	480
Next Breakpoint--Line at:	0

Hyphen; allows you to add hyphens as the formatter runs it will stop to allow insertation of hyphens.

NOTE!! The formatter handles only 40 colunn text and if the text file is a TI-Writer file, it is broken down to 40 col.

Jotter; the fromatter's own word processor. This also has a subaenu.

LoadF SaveF PrintF Edit Back Clear

Go; causes the formatter to create your master file for the Scheduler.

NOTE!! Be sure to name the files which are always displayed at the bottom of the screen. The printer default in all programs for TPA, which is PIO.CR, is Gemini compatible, and can be left as is. The Scheduler treats both picture and text files as graphics files. This is how it places everything just where you want it down to the exact dot on a page. TXTFILE is the file you are using as a text file. FNTFILE is the font that you have selected. EXTRNFILE is the file to be created for the Scheduler.

When entering the Scheduler be sure again to name the files displayed at the bottom of the screen. Use the same name for EXTRNFILE as you used with the Formatter to create the file and use a unique name for the SCHFILE, as this is the file that the Scheduler creates containing information on where to place the files on the screen as well as the size of the files in character dots, etc.

The Scheduler menu is as follows:

GO ModifyS ReadS WriteS ClearS

Directory eXit

Go; is again the last selection used.

ReadS; reads a previously created Schedule.

WriteS; writes the present Schedule to disk.

Directory; gives a disk directory.

ModifyS; allows you to set up the schedule with this menu.

Edit Up Down Insert Active

Print Size Blockmove Zap eXit

Up; advances to next schedule item.

Down; advances to previous schedule item.

Insert; allows insertion of text.

Active; prints schedule data on printer.

Size; sizes the individual file in dots to allow determination of the start and end of a file for selecting placement of the next file. You can select up to 75 files in a schedule for a page.

Blockmove; allows you to move blocks of text, etc.

Zap; removes an item from the schedule.

Edit; allows you to edit a schedule file or item, the following items are on the screen.

```
1 Filename:
Row:  #   #
Col:  #   #
#Reps:#
```

The number preceeding the filename is the position of that file in the file sequence of the schedule.

Filename; is the name of the item to use in that position.

Row; gives the starting and ending dotrow for the file.

Col; gives the start and end dotcolumn of the file. (You need only enter the starting values in Row & Col as when you use the size command after entering the starting values, the ending values are computed and listed to allow you to know where to start the next file in the schedule.)

#Reps; number of times to print the particular file.

When you first start to try to use The Printer's Apprentice, it appears to be very cumbersome and hard to use. But as you use it more and start to become familiar with all the aspects of the programs, you will find it fairly easy to use and the limitations of the system are only as limited as your imagination.

I hope that this article will help you to have as much fun with The Printer's Apprentice as I have been having, and I have only begun to explore the possibilities.

There is a second font disk available as well as a Toolbox disk available, and I have ordered mine. The Toolbox disk even has a program for creating borders so that you can create certificate type pages.

One thing that I really like about TPA is that its fonts are complete, including upper and lower case, numbers, and all the symbols.

==========

What's Next for the TI Community
By Dave Ramsey
From Manners Newsletter
Dec 87 - Jan 88

==========

Right now, there are lots of TI owners who are debating whether to stay with their current machine or to move on to something else. Now, that's a matter that has to be decided by you. Often it is decided for you, especially in the case of business usage. Sometimes what you need to do is supported with software available only on one machine. In such a case, your options are very small. In other cases, there is simply the desire to be part of the "mainstream" of microcomputing. In today's market, that means IBM PC compatible or perhaps using a Macintosh. But there is still that segment of the computer population that uses computers at home and for fun. There are a lot of those people in the 4A community. They are the folks I want to talk to here.

There's been a lot of static about the Myarc 9640 computer. Myarc has caused much of it by failing to deliver promised goods on schedule. A lot of it is due to disgruntled owners who don't yet have anything to run on their new computer besides MYWORD and TI-99/4A applications. I've yelled recently as loud as anyone and no one is even sure if the situation at Myarc is improving. But something else is starting to happen as well. Programming tools are starting to appear for the 9640. Excellent examples of MDOS oriented code are popping up. Documentation is beginning to appear.

Last night, without having ever written any serious MDOS code before this besides a few simple "Hello World" type programs, I wrote a file typing utility called MORE. MORE is designed to replace the MDOS TYPE command. If you've used MDOS or CPM or many other systems, you know that the TYPE command usually just types out a text file to the screen without pausing. Often it will scroll by so fast that it can't be read. You can usually stop it with CTRL-S but that is a haphazard operation at best. I wrote MORE to solve that problem. MORE was designed and written, assembled, and debugged in a matter of 6 hours total. Because of the excellent system interface designed by Paul Charlton, writing assembly programs for MDOS is very easy.

This brings me to the heart of the issue. Right now, the 9640 is capable of running almost all of your 4A

software. In a short while, the other creative people like you and me are going to start filling the software void in the MDOS world. For the hobbyist, this is all new territory. Who will write a terminal emulation program for the 9640? Who will write the first native mode programming editor or word processor for the 9640? Who's going to write the first game using 512 x 424 hi-res graphics? I'll tell you that I already have a relational database manager for my Z80 with its own ad hoc query language. I'm thinking real hard about how to port that to the 9640. Right now, it is fully as powerful as DBase-II was. With suitable changes, it could even be made DBase compatible.

I admit that there are advantages to owning a popular computer. My Z80 CPM system has public domain and commercial software coming out my ears. But there is, at least for me, a tremendous interest in new computers. There is a challenge in writing some of the first applications for a particular machine. There is a great deal of enjoyment in the comraderie that belonging to a close-knit group of hobbyists can bring.

Right now, the MDOS community has better tools at their disposal than the CPM community did when it began. The excellent TI Editor/Assembler can still be used from within the 99/4A emulation mode as can the MYWORD editor for both word processing and program development. Paul Charlton's linker provides the other critical tool for working with MDOS. With it, you can create a 9640 applications too. The system documentation, while still not voluminous, covers all the important points. With tools just like this, the CPM community created text editors. They wrote communications programs and true relational database managers. They wrote C and Pascal compilers as well as LISP interpreters. They wrote them all in assembler using tools even simpler than these.

I think that these kinds of challenges and opportunities are what many in the 4A community are really looking for. You have the added advantage that even though this is new territory, you don't need to learn another assembly language all over again. So let me say this - if you use a 4A purely as a hobbyist and if you are interested in the new 99xxx family of chips, think hard about getting a Myarc 9640 for yourself. If you thrive on challenges but want a better programming environment than the 99/4A currently offers, the 9640 is definitely one option. Consider it and then help other 9640 owners make this a fun and useful computer to own. As I said, it's a personal decision. But if programming fun is one of your considerations, the 9640 will fulfill it.

Editor's note: AMEN.....

==========

DISK CONTROLLERS
By Jerry Coffey
MANNERS NEWSLETTER
May - June 1987

==========

The disk capacity of the TI99 has increased in just a few short years from less than 80k (a single one-sided 35 track drive) to almost 2.9 megabytes (four double-sided, double density, 80 track drives). The early stand-alone was replaced by the PEBox system which would support three double-sided 40 track drives (540K). Corcomp introduced their four drive double-density system (1440K), followed by Myarc's similar system with two double-density formats (1280K and 1440K). Then in 1986, Myarc offered its 80 track upgrade which doubled capacity again. Even as capacity was increasing rapidly, the TI and Corcomp controllers differed only modestly in I/O speed. When Myarc introduced its fast DSDD controller, few reviewers did justice to its speed advantage. Early comparisons were done at the standard TI or Corcomp interlace, but the big speed gains required taking advantage of the much tighter sector interlace possible with the high-speed MYARC card. To understand how this works we need to take a look at the way a disk drive performs.

DISK DRIVE FUNDAMENTALS

A floppy disk drive writes information in concentric rings called "tracks" on a thin plastic disk coated with a film of magnetic particles. Each track in turn is divided into blocks of information called sectors. A blank disk has one (or more) index holes used to synchronize the process of writing to and reading from the disk. The type with many holes are called "hard sectored" since each sector has its position fixed by an index hole. The type of disks used by most computers have only one hole and are called "soft sectored". In this system the computer must write magnetic signposts on the disk to mark out each sector in a process called "formatting" or "initializing" a disk. These signposts take up a substantial fraction of the space on a track since they include not only sector numbers but buffers (filler bytes) that allow the computer to get into synchronization to read or write sectors of data and to prevent the sector identifier from being overwritten by a drive operating at a slightly different speed from the drive that formatted the disk.

The typical 5.25 inch disk drive has a "stepper motor" capable of moving the drive's read/write head(s) in or out along a radius of the disk in steps of 1/48 of an inch (thus the terminology "48 tpi" = 48 tracks per inch). Since the inner tracks have a smaller circumference, they crowd the bits of information

together. Magnetic coatings on a floppy disk are rated by their capacity in bits per inch at standard magnetic flux for the write head. This figure is usually over 5000 bpi for modern floppies, but was somewhat lower a few years ago. The circumference of the inner track of a 40 or 80 track disk is about 10 inches -- which allows about 6250 bytes to be writen on the track without exceeding 5000 bpi. For comparison, the Corcomp double density format requires over 6400 bytes per track. Media limitations were the reason that some early 5.25 disk drives only used the outer 35 tracks. The 16 sector (by 256 bytes/sector) format recommended by most drive makers requires only 6250 bytes per track and includes several hundred additional "buffer" bytes to compensate for differences in drive timing.

Timing is EVERYTHING

With soft-sectored disks, the integrity of the read/write processes require critical timing. The disk rotates at 300 rpm within a small margin. This means there are about 250 thousand magnetic pulses (bits) passing beneath the head each second. In single density format, the majority of these pulses are timing or filler bits -- in double density, many of the timing bits are suppressed in order to double the rate of data bits. In a typical sector read the drive must bring the disk up to speed, recognize the index hole, step out to track zero (to get its bearings), determine single or double density, verify its position, step in to the target track, verify the track number (written in the format operation), detect the sector identifier as it flies past, then immediately read the 256 data bytes into memory. Five of these operations require accurate reading of the magnetic pulses whizzing by at over 250K bits per second.

If you do some quick arithmetic (256 bytes/sector = 2048 bits/sector into 250K bits/second)... hmm... Why can't the drive read a 125 sector file in one second? Well first many of those bits are not data bits, they are overhead to keep things synchronized and allow for timing variation between drives. Second, some time is used moving the head from one track to the next when more than one track must be read. Third, 250K is the instantaneous read rate and the computer must take time to do other things like move the last sector out of its buffer to make room for the next one. In the standard TI protocol for reading a disk, the data is moved into VDP ram (so the drive could be used without the memory expansion) before it goes to the expansion memory. All this thrashing eats great chunks of the time available for reading data. By the time one sector is safely tucked away in the 32K card, several sectors have already passed by the drive's read head. If the sectors were written consecutively on the disk, we would have to wait a full revolution (0.2 seconds) before the next sector would pass under the head. To avoid this inefficiency, the consecutively numbered sectors are spaced out around the disk so that they are separated by just enough time to take care of other business. The actual pattern in which the sectors are scattered is called the "interlace". The idea of the interlace is to spread the sectors out to match the timing needs of the hardware -- both the time needed to stash each sector and the time needed to step from one track to the next and get the head settled down for some serious (250K bps) reading.

Interlace and Head Step Times

Life was simple with the TI disk controller. Both the interlace and the head step time were locked into the controller's PROM (that's the programmable chip that contains the control programs for the card). The head step time is the built-in delay between step signals to allow the stepper motor to move the head one "click" in or out. The TI settings are very conservative (read "slow") to allow for slow drives. The step time is 20 ms -- if you step from track zero to track 39, it takes 20x39=780ms, almost four revolutions of the drive. The TI interlace lays the sectors down on a track in the order 0753186942. This allows all sectors to be read in four revolutions of the disk though the slow head step lets another revolution go by between tracks. Thus the maximum read rate is about 9 sectors per five revolutions (=one second) or 2304 bytes per second.

When Corcomp designed its double density disk controller, allowances were made for the increased speed of later drives by permitting the step rate to be set with dip switches for each drive. The step rates available are 30, 20, 12, and 6ms (the faster values quoted in CC manual are referenced to the wrong clock speed). They also provided a choice of interlace options, though only a couple of them are practical. The default interlaces are labeled "7" for single density and "18" for double density. The single density interlace is the same as TI's, but with a faster step setting the head can be moved without losing a revolution and thus reads 20% faster than the TI controller. The double density interlace allows 18 sectors to be read in five revolutions, but it doesn't leave enough margin to stash the last track (that's why the sector number "hangs" for 0.2 seconds each 18 sectors while verifying a formatted disk -- you are seeing the extra revolution needed to acquire the first sector of the next track). Thus the maximum read rate is 18/1.2 or 15 sectors per second, about 47% faster than the TI controller. Users of the CC controller have probably noticed that it loads its own MANAGER program faster than this. In this case a special loader bypasses VDP and loads directly to CPU RAM -- this faster handling of the data alows the stepper motor to be activated sooner and saves one revolution per track (so the 98 sector file can be read in about 5.5 seconds).

This provided a foretaste of the speed that MYARC would achieve with its double density controller.

The MYARC controller bypasses VDP RAM to load directly to CPU RAM. This technique coupled with a buffer RAM chip on the controller card provided a quantum jump in disk I/O speed. The MYARC card reads the TI single density interlace at 11.25 sectors/second (the same as Corcomp) and reads the CC 18 sector/track interlace at 18 sectors/second (the same speed Corcomp reads its MANAGER program), but this is only the beginning. Since the hardware empties its sector buffer faster, consecutive sectors can be placed closer together allowing a track to be read in fewer revolutions, i.e., it supports a faster interlace. With fast drives, the 9 sector/track single density format can be read at an interlace of "2". (NOTE: In the MYARC terminology, the interlace number represents the number of disk revolutions required to read a track.) This works out to 22.5 sectors/second compared to 9 for the TI and 11.25 for the CC controller. The MYARC 16 sector format can be read at interlace "3", 26.67 sectors/second -- 3 times as fast as the TI controller and almost twice as fast as Corcomp double density. The Corcomp 18 sector format can be read at interlace "3" or "4", but the data rate is the same in either case, 22.5 sectors/second. Interlace "4" is smooth but requires a very quick head step, interlace "3" reads the track in 3 revolutions but forces an extra revolution for the step from track to track because sectors 17 and 0 are adjacent on the disk. Though both interlaces have the same data rate, interlace "3" is safer if you are uncertain about the speed of your stepper motor.

In order to read and write both double density formats, the MYARC system must insert an additional step in some I/O operations -- sector zero must be read to determine whether a double density disk has 16 or 18 sectors per track. This datum is needed to convert the logical sector numbers used by the TI operating system into track and sector-within-track addresses for the floppy disk controller chip. The TI and Corcomp controllers do not need this step because they do not use the full potential of the TI disk I/O protocol. Once this step, accessing sector zero, is added to the various disk operations, it opens the system up for using more than two formats -- including 80 track formats.

Beyond Double Density

A two format system can be managed using only the floppy controller's inherent ability to sense single and double density recording patterns. To get beyond this limitation, the additional data stored in sector zero must be read, stored, and used to modify the special binary commands sent to the FDC (floppy disk Controller) chip. Fortunately the TI99/4A system design already provides for such innovations through the Device Service Routine concept and standard "GPL" calls. The system doesn't care what hardware is attached as long as it plays by the rules -- an interface program stored in a memory chip (PROM) on the peripheral device does the trick. This program handles calls for I/O operations from other programs such as TI Writer or the Basic interpreters. Another set of rules controls the way disk and file information are saved on a disk. Disk parameters are stored in sector 0, while sector 1 must have a two byte "pointer" (a hexadecimal sector address) for each block (one sector) containing the bookkeeping data for a file. It is these blocks that are scanned in order to display the disk directory.

Since the MYARC controller must read sector zero to determine the number of sectors per track, the other parameters in that sector are available to control other variables such as number of tracks. But there were other limitations to overcome. The number of files on a disk is limited by the space available for pointers. 256 bytes at 2 bytes per pointer would give 128 files -- except the pointer list must end with a null word (>0000) so directory routines know where to stop -- so we get 127 files per disk. The pointer itself can address sector numbers as high as 65535, so this is no problem. The real limitation is the bit map in sector zero. It begins at byte 56 leaving only 200 bytes or 1600 bits available to map the disk. Since a bit must be turned on for each sector used, the 1440 sector DSDD 40 track disk is already near the limit. The answer devised for the 80 track DSDD system is to map two consecutive sectors within each bit. It wastes some space but no more than systems that use a standard 512 byte sector.

Making the Quad System Work

So now let's say we have new code in the disk controller EPROM (an "erasable" version of the PROM chip used by TI) that does all the proper tricks with the bit map and has the FDC commands to control the new 80 track drives we have added to the system. We still have to tell the controller which drives are 80 track and find a disk manager program that can use the new commands. The selection problem can be taken care of using the DIP switches on the card (but in the process you lose their original function -- setting step speed). Since the Eprom responds to standard GPL calls, most functions can be handled by the TI Disk Manager 2 cartridge. The exception is the disk formatting process -- the formatting works OK, but the initial data written into sector zero is for the standard bit map. (This can be fixed by changing byte 56 from >03 to >01 with a sector editor.) Read/Write operations from XB or TI-Writer work fine since they use the GPL protocols. Myarc has an excellent disk manager program that works beautifully with 40 track drives, but it has suffered from a number

of subtle bugs in 80 track mode. This program, like many others designed for high speed I/O, uses assembly language code to handle the FDC -- bypassing some of the routines in the EPROM. Differences in bit map handling, even slight differences in execution times can affect the performance of 80 track drives. The code in the 80 track EPROM has had a lot of attention to proper timing -- the price you pay for higher performance.

Fine Tuning the MYARC Disk System

Before you start using the Myarc system routinely, there are some experiments that can get you maximum performance from your drives. Use the Myarc disk manager to try different interlace settings -- first with your 40 track drives, then with the 80 track drives. Watch for hesitations as each formatted disk is verified, then use the Test option to read the sectors you have laid down. Look and listen for "retries" -- when the sector number pauses with a head seek noise. Use the best disks you have and note the combinations that test smoothly. With fast drives in good condition, you should be able to run 9 sector (single density) format at interlace 2 and 16 or 18 sector double density format at interlace 3. Don't worry if 18/3 pauses at the end of each track -- this is just the extra revolution forced by having sectors 17 and 0 adjacent on the disk.

When you try this with 80 track drives, don't be surprised if the results are different. The time required for the head to settle into a wide standard track may not be adequate to get it reading properly from the narrow tracks on the quad drive. Such subtleties as erase delays and disk quality are also more critical on the skinny, low power tracks. My Mitsubishi 4853s (96tpi) will support both 16/3 and 18/3 but are unreliable at 18/4, while my TEAC 55Bs support all three at 48 tpi. Don't take chances with any setup that is marginal. The error rate may be low, but it always seems to happen to a file that isn't backed up.

Hot Rodding

If you want to try for a little more speed, there are two more tricks you can use. The faster WD1772 FDC chip is pin compatible with the standard WD1770 supplied by Myarc. It will try to step the head at 2ms rather than the 6ms setting of the standard chip. (The 80 track EPROM automatically uses the fastest step speed available.) Many of the latest drives can step at 2ms or 3ms even though they are conservatively rated at 4ms or 5ms. The change is noticeable but may not be worth the high price of the WD1772 (it is not a commonly used chip and is rarely discounted). The second fix is cheap and very useful for producing large quantities of copies. The FDC chip's automatic "write verify" function can be defeated by shorting one pin on the controller card to ground. This is best done with a switch so the verify can be enabled for normal operations. The effect of this modification is equivalent to the "turbo" option on the Corcomp controller and should be used only after testing.

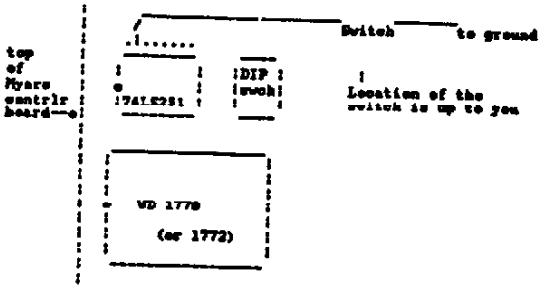
Interlace Patterns

NOTE: The configurations marked * and ** are the standard interlace patterns for the TI and Corcomp formats. The end-of-track intervals are only approximate since the 9 and 16 sector formats include more buffer space than the 18 sector format.

Sect/trk	Interlace	Pattern (dashed line is time available for head step)
9	4 *	0 7 5 3 1 8 6 4 2
9	2	0 5 1 6 2 7 3 8 4
16	5 **	0 11 6 0 15 1 12 5 9 16 2 13 6 10 17 3 14 7
16	4	0 9 5 14 1 10 6 15 2 11 7 16 3 12 8 17 4 13
16	3	0 6 12 1 7 13 2 8 13 3 9 15 4 10 16 5 11 17 mess
16	5	0 13 10 7 4 1 14 11 8 5 2 15 12 9 6 3
16	3	0 11 6 1 12 7 2 13 8 3 14 9 4 15 10 5

16-sector patterns are not precisely to scale

Here are the details for performing the "turbo" modification to lock out the "read after write" (write verify) usually performed by the controller: Find the 74LS251 chip at the top center of the controller board, above the dip switches and beside the large FDC chip (marked WD1770). Solder a wire from the number 2 pin of the 74LS251 through a switch to ground (e.g. the wide trace of the dip switches or any trace connected to that wide trace). It looks about like this from the bottom (non-component side) of the board.



As always you proceed at your own risk. (one person has told me this did not work on his 40 track system, but I haven't verified that.) You can tell it is working if your controller writes as fast as it reads (normally the write takes twice as long).

Since I wrote the article on disk controllers, I have discovered some surprising facts about my own system. All of the Myarc timings in the article were done on an 80 track system with the fast WD1772 controller chip (stepping at 2ms). With some help from Paul Charlton and Richard Roseen, I recently customized Paul's EPROM to step at 3ms using the WD1772. (Richard's drives were making errors at the faster speed.) I used a Mechatronics Eprom programmer to download the Eprom code to disk, changed the FDC commands with a sector editor, and wrote the altered code back to a fresh Eprom. The process is simple (and cheap) once you decide what code you need in the Eprom.

The slower step speed made it possible to notice some slight differences in the performance of the WD1772. The first thing I noticed was that interlace 4 on 18 sector tracks was no longer smooth -- it was missing the first sector after a track seek and forcing an extra revolution of the disk. this was the first clear indication of how close this format is to the "ragged edge". The reaction to the small change in step speed implies that this interlace comes within 5% of the minimum time required to step and settle the head. Thus the likelihood of read/write errors is relatively high with this interlace. It will occasionally detect the sector ID and begin to read or write before the head has completely settled. This interlace should definitely be avoided -- 18/3 is both faster and more reliable.

The Eprom modification itself was an interesting experience. I patched the new FDC commands into some unused text bytes and patched addresses into the code to point to the new locations. The Mechatronics Eprom programmer is an excellent piece of equipment. It will burn (program) a 2764 (8K) in about 90 seconds using the fast algorithm. I have talked to Jim Horn and Jeff Guide about offering an Eprom service to the customers of Disk Only Software. There are many possibilities this technique opens up. There is the 80 track modification for the TI controller worked out by Andy Cooper. Many Myarc owners are still using old Eproms that have never been upgraded (though this situation has improved since Lou Phillips increased his production capacity). The fix we developed for Richard's controller can provide the optimum step speed (3ms or 5ms) for different disk drives using the WD1772 FDC chip. Any enterprising programmer can get his tailor-made code installed in nonvolatile memory.

XXXXXXXXXXX

PRBASE ENHANCEMENT

By Bill Zaebst
Ft Cleveland TI Users Group

Note: The following looked interesting, and I decided to try it before putting it in the newsletter. I found a couple of problems and it took me quite some time to accomplish what looked to be a simple task. Number one, before you do anything, go in with DM1000 and "unprotect" the LOAD and PRBUTL/BAS programs or you cannot write to them after you make your changes. I tried just typing in the following as is and continually got an error message in line 290. The key may be "generated" in the REM statement. So, I loaded in the LOAD program from PRBASE, resequenced the lines to start at 100 and typed in the additional lines. I had to resequence a couple of times to get it to come out the same as shown. It ran without any problems. Also, since you are going to that much trouble, it would be nice that when you finished with the PRBUTL/BAS you could go back to the LOAD program if you so wished to get into PRBASE. On mine at line 250, I changed the reference to line 1230 to line 1229. I typed on line 1229, RUN "DSK1.LOAD" and it returns to the load program. You could insert a CALL KEY which would allow you to either quit or go to LOAD as you wish. Also, if it is your intention to go to the Utility Program instead of loading PRBASE, be sure to have your hand on the space bar as the LOAD program commences.

PRBASE is a nice program so nothing drastic needs changing. But since copy utilities were not included orginally but added later in the form of an Extended Basic program. Nothing is provided to automatically load that part of the program.

The main menu is in assembly and cannot load XBASIC. However, the LOAD program that loads the assembly can be used to load the copy utilities before you get to the main menu. Another menu could be provided, but it is simpler to use a CALL KEY to detect depression of the space bar and as a result automatically load the copy utilities. Since the load program did not previously display anything, it has been changed to provide screen color to match the balance of the program and to display the loading message, as it takes about fifteen seconds to load.

```
100 !$$$$$$$$$$$$$$$$$$$$$$$$
110 !$                        $
120 !$    Generated By        $
130 !$    Systex V1.0         $
140 !$      (C) 1985          $
150 !$   By Barry Boone       $
160 !$                        $
170 !$  Hold Space Bar To     $
```

```
180 !!  Load Copy Utility  !!
190 !!                     !!
200 !!!!!!!!!!!!!!!!!!!!!!!!
210 CALL KEY(0,K,S):: IF S=0 THEN 270
220 FOR I=1 TO 8 :: CALL SCREEN(5)
230 CALL COLOR(I,16,1):: NEXT I
240 DISPLAY AT(2,4):"PRBASE COPY UTILITIES"
250 DISPLAY AT(6,4):"LOADING, PLEASE WAIT"
260 IF K=32 THEN RUN "DSK1.PRBUTL/BAS"
270 CALL INIT :: CALL LOAD(8196,254,0)
280 CALL LINK("SLOAD")
290 CALL LINK("ENTRY")
```

The above program inserts the word copy in the title to more accurately describe it's function. Line 150 of the PRBUTL/BAS program should also have copy inserted and have column number adjusted accordingly.

In order for the title not to be cleared, the CALL CLEAR in line 100 of PRBUTL/BAS must be changed to a DISPLAY AT():"" (NOTHING to only clear the one line on which the loading message is displayed.

Go to PRBUTL/BAS and change the following:

```
100 CALL LINK("CHARDF"):: CALL SCREEN(5):: DISPLAY
    AT(6,1):"" :: FOR A=0 TO 14 :: CALL COLOR(A,16,1)::
    NEXT A :: OPTION BASE 1 :: ON ERROR 1280
150 K=1 :: L=2 :: DISPLAY AT(2,4)ERASE ALL:"PRBASE COPY
    UTILITIES" :: CALL HCHAR(4,1,45,32):: CALL
    HCHAR(21,45,32)
```

<()><()><()><()><()><()><()><()><()><()><()><()>

POSTER CONTEST

SUBMIT BEFORE

JULY MEETING !

WIN A BOX OF

PRINTER PAPER

HAM AND CHIPS

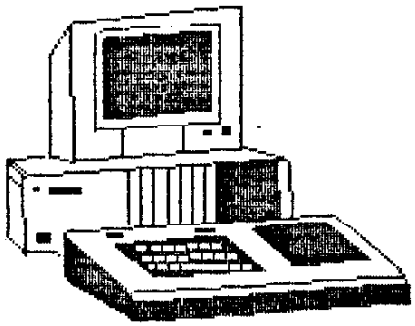
FLEA MARKET

AUGUST 20, 1988

WE WILL HAVE A

TABLE FOR TI GOODIES

BUY - SELL - SWAP



TIPS FROM THE TIGERCUB

#44

TIGERCUB SOFTWARE
156 Collingwood Ave.
Columbus, OH 43213

Distributed by Tigercub Software to TI-99/4A Users Groups for promotional purposes and in exchange for their newsletters. May be reprinted by non-profit users groups, with credit to Tigercub Software.

Over 130 original programs in Basic and Extended Basic, available on cassette or disk, now reduced to just $1.00 each, plus $1.50 per order for cassette or disk and PP&M. Cassette programs will not be available after my present stock of blanks is exhausted.
Descriptive catalogs, while they last, $1.00 which is deductable from your first order.

Tigercub Full Disk Collections, reduced to $5 postpaid. Each of these contains either 5 or 6 of my regular $2 catalog programs, and the remaining disk space has been filled with some of the best public domain programs of the same category. I am NOT selling public domain programs - they are a free bonus!
TIGERCUB'S BEST, PROGRAMMING TUTOR, PROGRAMMER'S UTILITIES, BRAIN GAMES, BRAIN TEASERS, BRAIN BUSTERS!, MANEUVERING GAMES, ACTION REFLEX AND CONCENTRATION, TWO-PLAYER GAMES, KID'S GAMES, MORE GAMES, WORD GAMES, ELEMENTARY MATH, MIDDLE/HIGH SCHOOL MATH, VOCABULARY AND READING, MUSICAL EDUCATION, KALEIDOSCOPES AND DISPLAYS

NUTS & BOLTS (No. 1), a full disk of 100 Extended Basic utility subprograms in merge format, ready to merge into your own programs. Plus the Tigercub Menuloader, a tutorial on using subprograms, and 5 pages of documentation with an example of the use of each subprogram. Reduced to $15.00 postpaid.
NUTS & BOLTS NO. 2, another full disk of 108 utility subprograms in merge format, all new and fully compatible with the last, and with 10 pages of documentation and examples. Also $15 postpaid.

```
*****************************
* NUTS & BOLTS #3 is now    *
* ready, another full disk  *
* of 140 new merge-format   *
* utility subprograms, all  *
* compatible with the pre-  *
* vious. With 11 pages of   *
* documentation, $15 ppd.   *
*****************************
```

TIPS FROM THE TIGERCUB, a full disk containing the complete contents of this newsletter Nos. 1 through 14, 50 original programs and files, reduced to $10 ppd.
TIPS FROM THE TIGERCUB VOL. 2, another diskfull, complete contents of Nos. 15 through 24, over 60 files and programs, also just $10
TIPS FROM THE TIGERCUB VOL. 3, another 62 programs, tips and routines from Nos. 25 through 32, $10 postpaid.
TIPS FROM THE TIGERCUB VOL. 4, another 48 programs and files from issues 33 through 41, also $10 postpaid.

Thanks to Steve Chapman and Bill Wallbank of Stone & Webster Engineering Corp. TIUG for this one. If V=21 you are in Extended Basic, otherwise you are in Basic. I am not sure it will work with all consoles and modules. -

```
100 RANDOMIZE (0)
110 V=INT(RND*100)
```

How can you input a blank (CHR$ 32) with ACCEPT AT? As far as I know, you can't. With LINPUT, just hit the space bar, and with INPUT, type " ". But with ACCEPT AT the space bar gives a null string and " " gives " " ! However, you can code around it -

```
X$=CHR$(34)&CHR$(32)&CHR$(32
):: ACCEPT AT(1,1):T$ :: IF
T$=X$ THEN T$=CHR$(32)
```

And, to clear up the puzzling behavior of the "quote marks" -

```
100 CALL CHARPAT(34,CH$):: C
ALL CHAR(35,CH$)!written by
Jim Peterson
110 DISPLAY AT(1,7)ERASE ALL
:"THE # PUZZLE":" You can't
enter PRINT # or PRINT ### -
 the computer  demands an
even number of #."
120 DISPLAY AT(5,1):"1 PRINT
 ## !prints a null string (n
othing)":"2 PRINT #*# !print
s *"
130 DISPLAY AT(8,1):"3 PRINT
 #### !prints #":"4 PRINT ##
### !crashes as  STRING-NUM
BER MISMATCH"
140 DISPLAY AT(11,1):"5 PRIN
T ###### !crashes as  SYNTAX
 ERROR"
150 DISPLAY AT(13,1):"6 PRIN
T ###### !prints ##":"7 PRIN
T ######## !prints ###":"8 PR
INT ######### !print ####"
160 DISPLAY AT(16,1):"9 PRIN
T ######### !prints ###":"10
PRINT ########## !crashes as
STRING-NUMBER MISMATCH"
170 DISPLAY AT(19,1):"11 PRI
NT ########### !crashes as SY
NTAX ERROR":"12 PRINT ######
#### !####"
```

```
180 DISPLAY AT(22,1):"13 PRI
NT ############ !#####":"14 P
RINT #############!######"
190 DISPLAY AT(24,1):"TRY IT
! LINE NO.(1-14)?" :: ACCEPT
 AT(24,25)VALIDATE(DIGIT)SIZ
E(2)BEEP:LN :: IF LN<1 OR LN
>14 THEN 190
200 CALL CLEAR :: ON LN GOSU
B 230,240,250,260,280,290,30
0,310,320,330,340,350,360,37
0
210 PRINT :;:;:"Press any ke
y"
220 CALL KEY(0,K,S):: IF S=0
 THEN 220 ELSE 110
230 PRINT "" :: RETURN
240 PRINT "*" :: RETURN
250 PRINT """" :: RETURN
260 PRINT ""*"" !crashes as
STRING-NUMBER MISMATCH - the
 * is misinterpreted as a mu
ltiplier!Same with +,-,/
270 !with anything else, inc
luding numerals, crashes as
SYNTAX ERROR - but inserts a
 space before the character!
280 PRINT ""**"" :: !crashes
290 PRINT """""" :: RETURN
300 PRINT """*""" :: RETURN
310 PRINT """**""" :: RETURN
320 PRINT """""""" :: RETURN
330 PRINT """"*"""" !crash
340 PRINT """"**"""" !crash
350 PRINT "":: RETURN
360 PRINT """""*""""" :: RET
URN
370 PRINT """""**""""" :: RE
TURN
```

The method of closing an "ajar" file, described in Tips #28, doesn't always work, but this one seems to be reliable -

```
100 ON ERROR 500 :: OPEN #1:
"DSK1.TEST" :: INPUT #1:A$ :
: PRINT A$ :: STOP
500 ON ERROR 510 :: CLOSE #1
510 INPUT "CHECK DISK AND DR
IVE, PRESS ANY KEY":DUMMY$ :
: RETURN 100
```

This one is just for the fun of it - it uses the contents of computer memory to create designs -

```
100 DISPLAY AT(3,10)ERASE AL
L:"COLORPEEK": :TAB(7);"by J
im Peterson": : :" Watch the
 computer's memory": :"displ
ayed in color."
110 DISPLAY AT(12,1):"Choose
": :"(1) plain colors": :"(2
) bars & checks": :"(3) patt
erns" :: ACCEPT AT(12,8)VALI
DATE("123")SIZE(1):Q :: CALL
 CLEAR :: IF Q=1 THEN 170
120 DISPLAY AT(12,5):"wait,
please" :: IF Q=3 THEN 140
130 FOR CH=32 TO 143 :: CALL
 CHAR(CH,RPT$("F0",8)):: NEX
T CH :: GOTO 160
140 RANDOMIZE :: FOR CH=32 T
O 88 :: FOR J=1 TO 4 :: X$=S
EG$("0018243C425A667E8199A5B
DC3DBE7FF",INT(16*RND+1)*2-1
,2):: B$=B$&X$ :: C$=X$&C$ :
: NEXT J :: CALL CHAR(CH,B$&
C$)
150 CALL CHAR(CH+55,B$&C$)::
 B$,C$="" :: NEXT CH
160 FOR SET=0 TO 14 :: CALL
COLOR(SET,SET+1,16-SET):: NE
XT SET :: CALL SCREEN(2):: G
OTO 180
170 FOR SET=0 TO 14 :: CALL
COLOR(SET,SET+2,SET+2):: NEX
T SET :: CALL SCREEN(16)
180 FOR J=-1 TO -2000 STEP -
1 :: CALL PEEK(J,A):: A=A-(A
<33)*(A+32):: A=A+(A>143)*(A
/2):: R=R+1+(R=24)*24 :: CAL
L HCHAR(R,1,A,32)
190 C=C+1+(C=32)*32 :: CALL
VCHAR(1,C,A,24):: NEXT J ::
GOTO 100
```

Unlike most of the number games played against the computer, you can win this one -

```
100 CALL CLEAR :: CALL SCREE
N(16):: DISPLAY AT(3,8):"THE
 '37' GAME" !by Jim Peterson
110 DISPLAY AT(5,1):" We wil
l take turns picking":"a num
ber from 1 to 5, but":"not t
he number that was just":"pi
cked."
120 DISPLAY AT(10,1):" The n
umbers we pick will be":"add
ed to the total count."
130 DISPLAY AT(13,1):" Whoev
er reaches 37 is the":"winne
r, but if you go over":"37 y
ou lose."
140 CALL SHOW(20,1,"Press an
y key to start")
150 CALL KEY(0,K,S):: IF S=0
 THEN 150
160 DATA 4,11,17,24,30,37
170 DATA 262,330,392,523,523
180 DATA 1047,784,659,523,52
3
190 C,P=0 :: CALL CLEAR :: C
ALL MAGNIFY(2):: R=10 :: FOR
 J=1 TO 5 :: CALL SPRITE(#J,
48+J,5,R,10):: R=R+30 :: NEX
T J
200 CALL SHOW(24,1,"(Y)ou or
 (C)omputer first?"):: ACCEP
T AT(24,28)VALIDATE("YC")SIZ
E(1):Q$ :: DISPLAY AT(24,1):
""
210 IF Q$="C" THEN CALL SHOW
(22,8,"1 pick 4"):: CALL COL
OR(#4,1):: P=4 :: C=4 :: CAL
L SHOW(3,10,"COUNT=4")
220 CALL SHOW(20,8,"Pick you
r number"):: ACCEPT AT(20,26
)VALIDATE("12345"):N :: IF N
=P THEN 220
230 IF P>0 THEN CALL COLOR(#
P,5)
240 CALL COLOR(#N,1):: P=N :
: C=C+N :: CALL SHOW(3,10,"C
OUNT= "&STR$(C)):: IF C=37 T
HEN 320 ELSE IF C>37 THEN 34
0
250 RESTORE 160
260 READ X :: IF C<X THEN B=
X-C ELSE IF X<37 THEN 260
270 CALL SHOW(22,8,"I'm thin
king..."):: FOR Y=1 TO 700 :
: NEXT Y
280 IF B>5 AND B/2=INT(B/2)T
HEN B=B/2
290 IF B>5 OR B=P THEN B=1-(
P=1)
300 CALL SHOW(22,8,"I pick "
&STR$(B)):: CALL COLOR(#P,5)
:: CALL COLOR(#B,1):: P=B ::
 C=C+B :: CALL SHOW(3,10,"CO
UNT= "&STR$(C))
310 IF C=37 THEN 340 ELSE IF
 C>37 THEN 320 ELSE 220
320 RESTORE 170 :: FOR J=1 T
```

```
0 5 :: READ F :: CALL SOUND(
100,F,5,F*1.03,5):: NEXT J :
: CALL SHOW(12,8,"YOU WIN!")
330 CALL SHOW(15,8,"Play aga
in? (Y/N)"):: ACCEPT AT(15,2
6)VALIDATE("YN"):Q$ :: IF Q$
="N" THEN STOP ELSE 190
340 RESTORE 180 :: FOR J=1 T
O 5 :: READ F :: CALL SOUND(
300,30000,30,30000,30,F,30,-
4,5):: NEXT J :: CALL SHOW(1
2,8,"YOU LOSE!"):: GOTO 330
350 SUB SHOW(R,C,T$):: FOR J
=1 TO 10 :: DISPLAY AT(R,C):
" " :: DISPLAY AT(R,C):T$ ::
 NEXT J :: SUBEND
```

A couple more peculiarities of the computer -

```
100 DISPLAY AT(3,8)ERASE ALL
:"POS PUZZLE #1": :"        f
rom Tigercub"
110 DISPLAY AT(9,1):"Why doe
s the computer say":"that X=
1 if you answer the":"prompt
 with the Enter key":"(null-
string) ?"
120 DISPLAY AT(14,1):"110 IN
PUT M$"
130 DISPLAY AT(15,1):"120 X=
POS(""TESTING"",M$,1)::":"PR
INT X :: GOTO 100"
140 !POS PUZZLE #1 - why doe
s the computer say that X=1
if you answer the prompt wit
h Enter (null-string) ?
 - Jim Peterson
150 INPUT M$
160 X=POS("TESTING",M$,1)::
PRINT X :: GOTO 140
```

And -

```
100 DISPLAY AT(3,8)ERASE ALL
:"POS PUZZLE #2": :"        f
rom Tigercub"
110 DISPLAY AT(7,1):"Why doe
s the computer say":"that th
e first position of":"null-s
tring is at whatever":"posit
ion it is told to start":"se
arch at?"
120 DISPLAY AT(13,1):"100 M$
=""""""
130 DISPLAY AT(14,1):"110 DI
SPLAY AT(20,1):""POS?"" :: A
CCEPT AT(20,6):P"
140 DISPLAY AT(16,1):"120 X=
POS(""TESTING"",M$,P):: DISP
LAY AT(22,1):""X="";X :: GOT
O 110"
150 M$=""
160 DISPLAY AT(21,1):"POS?"
:: ACCEPT AT(21,6):P
170 X=POS("TESTING",M$,P)::
DISPLAY AT(23,1):"X=";X :: G
OTO 160
```

Here is an improvement to the PRINTSPEAKER in Tips #40 - in lines 130 and 160, change the CHR$(1)&"1" to CHR$(3)&"255" . This will avoid problems if the program being converted opens FILE #1.

Irwin Hott informs me that assembly routines which have been imbedded into XBasic programs, using ALSAVE or SYSTEX, can be saved to cassette and reloaded. This could be very useful for those who have a stand-alone or "matchbox" 32k.

And, a mini-game for you to have fun with or improve on -

```
1 !      2-LINE GAME
       by Jim Peterson
- use S&D keys to paint the
white line on the highway
2 !if it is too easy, change
the 6 in A$=RPT$(CHR$(143),6
) to 5 and the 5 in C>T+5 to
 4
100 CALL CLEAR :: A$=RPT$(CH
R$(143),6):: CALL COLOR(14,2
,2,2,16,16):: CALL SCREEN(4)
:: T=11 :: C=14 :: CALL HCHA
R(22,C+2,42):: RANDOMIZE
110 T=T+INT(3*RND-1)+(T=21)-
(T=1):: PRINT TAB(T);A$ :: C
ALL KEY(3,K,S):: C=C+(K=83)-
(K=68):: CALL HCHAR(22,C+2,4
2):: IF C<T OR C>T+5 THEN ST
OP ELSE 110
```

And finally, one of the best examples of compact programming I have ever seen -

```
1 !JOHN WITTE'S 3-LINE VERSI
ON OF JOHN WILLFORTH'S WAVE
POWER - PUBLISHED IN GREATER
 OMAHA UG NEWSLETTER
100 CALL CLEAR :: A$(1)="ABC
DEFGFEDCBA" :: FOR I=1 TO 7
:: CALL CHAR(72-I,RPT$("0",2
*I-2)&"FFFF",47,"30303EFF7F3
E1E04"):: A$(I+1)=SEG$(A$(I)
,2,12)&SEG$(A$(I),2,1):: NEX
T I
110 CALL SPRITE(#5,47,2,180,
180,-23,0,#6,47,2,80,100,-23
,0):: CALL MAGNIFY(2)
120 FOR I=1 TO 12 :: PRINT A
$(I+(I>7)*2*(I-7))&A$(1+I+(I
>6)*2*(I-6)):: NEXT I :: GOT
O 120
```

Memory full
Jim Peterson

YDC MEETING

JUNE 11, 1988

9:45 AM

GENEVE OPERATION

PRINTERS APPRENTICE

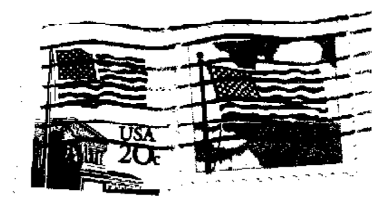
PROGRAM LIBRARY

MULTIPLAN

TI-WRITER

PHILADELPHIA AREA
TI-99/4A USER'S GROUP
c/o DON ARSENAULT
1290 BUTTONWOOD DR.
LANSDALE, PA 19446

SOUTHEASTERN, PA JUN 13 1988



FIRST CLASS MAIL

DALLAS TI HOME COMPUTER GROUP

P.O Box 29863
DALLAS, TX
75229