

-The Front Ranger-



The Newsletter of the Front Range 99er Computer Club

Vol. 3 No. 4

P.O. BOX 9572, COLORADO SPRINGS, CO 80932

Jan. 1986

PRESIDENT.....	BONNIE SNYDER.....	578-1583
VICE PRESIDENT.....	KEVIN KAPUS.....	576-3199
SECRETARY.....	BRYAN HALL.....	548-1277
TREASURER.....	BRYAN MINNIGERODE.....	599-8506
EDITOR.....	KEVIN KAPUS.....	576-3199
LIBRARIAN.[program].....	JOE NUVOLINI.....	596-6938
LIBRARIAN.[lending].....	GENE FLEMMING.....	635-8162
MEMBERSHIP.....	WARREN WIEKHORST.....	596-7724
PROGRAM.....	STEVE FUNKHOUSER.....	633-3007
PUBLICITY.....	CLIFF SWANK.....	473-1101
ASSEMBLER SIG.....	STEVE JACOBS.....	594-9523
T.I. WRITER.....	JOE NUVOLINI.....	596-6938
T.I.FORTH.....	NATHAN RACKOFF.....	598-1098
COMPUSERVE SIG.....	KEVIN KAPUS.....	576-3199
TIBBS.....	BULLETIN BOARD.....	574-5762
VILLA-TI.[library].....	BULLETIN BOARD.....	574-2567

JANUARY 1986						
S	M	T	W	T	F	S
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

***** NEXT MEETING

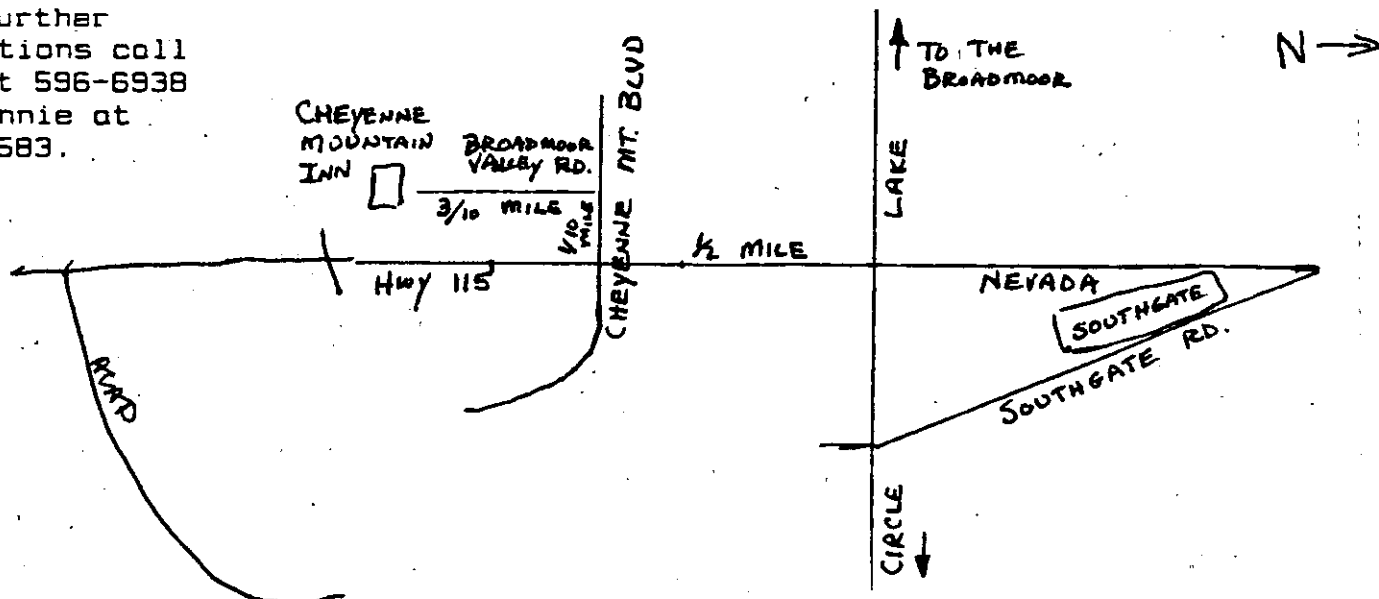
DATE.....THURSDAY, JANUARY 16, 1986

TIME.....6:30 PM

PLACE.....THE CHEYENNE MOUNTAIN INN, ON BROADMOOR VALLEY ROAD

PROGRAM.....The program for January will include demonstrations of 4A/TALK, Fast Term, TE2, TE3 and a demonstration of CompuServe.

For further
directions call
Joe at 596-6938
or Bonnie at
578-1583.



Any letter or comment to the Editor should be sent to :Front Range 99er Club
P.O. Box 9572 Colorado Springs, CO. 80932

The Presidents Corner

Happy New Year to you all. Here's hoping that 1986 is the best year ever for each of us personally, and for the Front Range 99'ers Computer Club!

For a moment, I would like you to remember back to July of 1985. That was the month that we all discovered Disk Manager 1000. Remember the message traffic on TIBBS touting DM1000 as the best discovery since white bread? How many of you obtained a copy of it either from our librarian at his home or at the July and August meetings or even after that? How many of you also obtained the updated version offered on TIBBS and VILLA? Well, the program is everything it is cracked up to be and much, much more. HOWEVER, although it was not documented correctly on the original version, DM1000 is FREeware written by Bruce Caron, a member of the Ottawa TI Users' Group and owned by that group. I would very much like to see us send the Ottawa group some money for this program. After all, if they and their member Bruce Caron can produce such class material as DM1000, surely they can supply us with other superior programs in the future, IF we pay the piper. For those of you who would like to facilitate matters, we will be taking up a collection (the amount is strictly up to you...we will not demand any particular amount) at the January meeting. We will then place the money in the FR99CC treasury and issue a check in that amount to the Ottawa Users' Group and mail it to them. You may also come by my home and leave your donation (up to the end of January) and it will be included in the check. Please remember that FREeware is the lifeblood of the TI99/4A software and it needs to be encouraged to continue, particularly when it is of the quality of DM1000! If you have any questions or comments about this project, please do not hesitate to call me and talk to me about it. By the way, if you do not wish to contribute to the general collection, you can still send your own private donation to:

The Ottawa TI-99/4 Users Group

PO Box 2144 Station D
Ottawa Ontario Canada K1P-5W3

Hope to see you at the up coming meeting. Bonnie

LARRY'S CLOCK SHOP

TREASURER'S REPORT

For Dec. 1 thru Dec. 31, 1985

CASH ON HAND, BEGIN \$ 1,694.70

Receipts 65.00

TOTAL CASH AVAILABLE 1,759.70

Disbursements (136.25)

CASH ON HAND, END \$ 1,623.45

Submitted:
J.H.B. Minnigerode
Treasurer

**CLOCK'S - WATCHES
SALES - SERVICE
HOUSE CALLS MADE**

10% Discount off retail price of all merchandise.

Hours
Tues. - Fri.
9:00 - 5:00

Saturdays
9:00 - 2:00

Phone
632-5959

722 E. Bijou
The corner of
Bijou & Union

(Offer good 1-2-86 thru 1-31-86 with ad.)

The following is a copy of a letter from Mr. Bob Boone of the Ottawa users group in Canada to Tex-Comp.

Bob Boone.
25 Ottawa St..
Arnrior. Ontario.
Canada K7S 1W7
October 26, 1985

Marketing Director
TEX-COMP
P O Box 33084
Granada Hills, CA
U.S.A. 91344

Dear Sirs,

I am writing on behalf of the Ottawa TI-99/4A User Group to express our extreme displeasure at your actions with regard to our 'Freeware' package; DM-1000.

In particular, we wonder why you felt you had the right to use it as a promotional item for your 'Explorer' package and worse: alter the LOAD file to promote yourselves as distributors of our software and to remove our FREEWARE notice!

Discovering that you were offering DM-1000 source code for sale at \$7.50 per copy was the last straw!

We purchased the source code from Bruce in May of this year and began distributing it (and the object code) as freeware in June. It is legally ours. We feel, at the very least, you were negligent in not contacting us before taking the steps outlined above.

A copy of this letter is also being sent to all known publications dealing with the TI-99/4A, with a covering letter indicating our User Group's point of view in this matter.

We haven't the desire or the resources to go to court over this matter but we can and will create a furor in the international TI world if steps aren't taken to rectify the above wrongs. Donations for DM-1000 have averaged \$10-\$15 dollars per respondent and as Terrie Masters of Beverly Hills California says: "we're not likely to see any return from those you've given away as a freebie". Please be good enough to publish a retraction of your offer of a "free copy of Disk Manager 1000" and publish our User Group's address as the sole source and beneficiary for DM-1000. An apology wouldn't hurt too!

Sincerely,

Bob Boone,
Past President
Ottawa TI-99/4A U.G.
Box 2144 Station D
Ottawa, Ontario
Canada K1P 5W3

Let's make sure that the members of this club compensate the Ottawa U.G. for this fine program. I suggest that we take up a collection, at the next club meeting, since most of our members are using this program and while we unwittingly distributed it as a public domain program, we now know the truth of the matter. A few bucks from each user of the program in our group seems appropriate!

More on Fast Term... by Joe Nuvolini

While playing around with Fast Term today I discovered that you can prewrite and send messages with Fast Term just as you can with 4A/Talk. The procedure is as follows:

1. Place a data disk in drive #1 and enter the file name of the message. Enter FCIN B to do this. Then press ENTER to open the file on the data disk. Next enter FCIN SHIFT D to get into half duplex. Now you can write the message on the screen. At the end of each line you must press ENTER and FCIN X to start the next line. When done with your message you press FCIN B again and save the message to disk. Now press FCIN SHIFT D again to return to full duplex before signing on to the BBS.

2. Once on the BBS select FCIN N and change the filename to whatever you named your message file. After you enter the heading portion of the message and get to the text portion, you enter FCIN ,. You are then asked if you want to send one line at a time. With TIBBS and the Villa answer Y. Then you press the space bar to enter each line. When finished, press enter and save the message. If you are uploading your message to a system that will take block uploads such as IBBS you answer N to the line at a time question.

That's it. Hope you all find this helpful.

Speech in the Box

Three members of the Front Range 99'er Computer Club have developed an interface card called "Speech-in-the-Box". The prototype was demonstrated at the December users group meeting. The card allows you to move your TI Speech Synthesizer into your Peripheral Expansion Box thus eliminating the two extra connections that cause so much trouble where the P-Box umbilical plugs into the computer console. The unit takes only minutes to install. You merely remove the speech card from the Speech Synthesizer, plug it into the Speech-in-the-Box card, and then plug the latter into one of the available slots in your P-Box. The unit will sell for under \$30 and will be marketed nationally by RMJ and Associates Home Computer Services starting in the first quarter of 1986.

ASSEMBLY AUTORUN

by N. A. Molander
Suncoast Beeper Newsletter

Many Assembly Language programs are written so that they

automatically execute when loaded. In assembly language source code this is accomplished by entering the program name in the operand field after the END statement. [pg 234 E/A manual] and assembling the program. The program would then execute, beginning at the memory address associated with the Program Name Label. The program name is the same as used in the E/A menu option 3 [LoadRun] or option 4 [Run]. The reverse is also true that by deleting this operand from the END statement in the source code and reassembling the program, it then will not automatically run when loaded. The pros and cons of this feature include:

AUTORUN EXECUTION SIMPLIFICATION
EASE OF LOADING MULTIPLE PROGRAMS
AUTORUN DEBUGGING INHIBITED

When source code is not available, object code can still be edited to alter this feature. The following assembly language source code program will be used to illustrate how to edit this feature in UNCOMPRESSED object code only:

```
DEF TEST
REF UMBW
TX TEXT 'TRIAL'
TEST LI R0,365
LI R1,TX
LI R2,5
BLWP @UMBW
JMP TEST
END TEST
```

Assembling the above source code with the E/A using the normal [uncompressed] option produces the following object code file.

00018	A0000B5452B44941B4C00B0200B00	
16DB0201C0000B0202B00057F35AF		0001
A0012B0420B0000B10F77FB9BF		0002
200067FED1F		0003
50006TEST 30014UMBW 7FADAF		0004
: 99/4 AS		0005

The last record of any object code file [line no. 0005] begins with a colon[:] and is followed by the 99/4 AS identification code [pg 238 E/A manual]. Preceding this are the program's external REFERENCES DEFINITIONS [line no. 0004].

The AUTORUN feature is contained in the preceding record [line no. 0003]. This sequence of records is identical for all object codes in either format.

The AUTORUN record format is: TXXX7YYYY. The record will contain no other information except line number.

The T tag designates that XXXX is the entry point of the program [see pgs 238, 307 309 E/A manual for further info]. The 7 tag designates that YYYY is the checksum [pg. 239 E/A] for the object code line. F is the end of record marker.

In the above example the relocatable Entry Point is >0006 [line 3 or 5]. If the program was loaded beginning at >A000 then the actual memory address of the Entry Point is >A000 + >0006 = >A006.

The checksum, >FED1, is the two's complement of 303 arrived at by adding the ASCII value of each character in 2006E7. (2=>32 [50] + 0=>30 [48] + 0=>30 [48] + 6=>36 [54] + 7=>37 [55] = 303)

Line 4 is the program's REF/DEF record. Tag 5 designates that

0006TEST is a DEFINITION with a relocatable address of 0006 and a symbol of TEST. If line 3 is eliminated, then the program can be executed by using the Program Name 'TEST'.

DELETING AUTORUN

To eliminate auto-execution delete the AUTORUN line from the object code file and resave the file. To execute this saved program will now require the Program Name, the symbol in the REF/DEF record with the same memory address. Autorun can be restored by re-entering the record in the object code exactly at the same line number and with the same format that was deleted.

These changes can be made with the EDITOR of the E/A program. As the procedure being described here only applies to UNCOMPRESSED object code files, be sure the format complies. The screen should be filled with ALPHANUMERICS, if there are only two spaces between object code tags (B's etc) and most are blank except for recognizable text, it is a COMPRESSED object code program and cannot be altered. Scroll to the last lines of the program to see the line sequence described above and proceed with the changes. DISKO, TI-WRITER, and DISKFIXER can also be used in the same way.

ADDING AUTORUN

Adding Autorun to an object code program is similar to replacing a deleted one except that the Entry Point desired must be chosen and the checksum calculated. The checksum will be ignored if the TAG is changed from a 7 to an 8. (E/A pg 241).

The most likely choice is the address of the Program Name used to run the program. In the above example by selecting 50006TEST as the entry address then line 3 would be created and inserted into the object code file as shown. By ignoring the checksum record 2000680000F could be entered as the record of line 3.

Of the 14 Object Code Tags used by the TMS9900, 5 relate to Absolute addresses and 5 to Relocatable addresses, (pg 240 E/A). Programs with absolute addresses use the ADRG directive and are placed in a specific memory location. The only effect on the above example would be that TAGS 2 5 become 1 6.

The simplified procedure presented here will work in many cases, however, complications can arise.

Relative Files Part 3 By John Pearce

In this installment we will look at a method for keeping track of data records using a mathematically calculated relative record number. Did you study modulus arithmetic as I suggested?

Let's talk about modulus arithmetic for just a moment. The modulus number is determined by dividing the given number by the modulus number and determining the remainder. The modulus 100 value for 1, 101, 201, 1001, and 5001 is 1. This feature of modulus arithmetic can be used to determine the relative record number based on some other number.

Let me use the TIBBS board as an example. Those of you who call TIBBS have probably noticed that TIBBS never has more than 100 messages

even though the message numbers are currently in the 3,000 range. That is a design limit by the author of TIBBS, Ralph Fowler. Fowler designed the message system to operate on relative file access based on modulus 100 of the message number. If you ask TIBBS to read a message, it calculates the relative record number by dividing the message number by 100. The actual statements in the program look something like this:

```
VA = QA - INT(QA / 100) * 100 IF VA = 0 THEN VA = 100
```

The variable QA is the message number to be read and VA is the relative record number which corresponds to that location. For example, to find message 3527 the value of VA from above is 27. Reading relative record 27 from the file will produce the text associated with message 3527.

Granted, this is a very simple example, but you should be able to see how this can be used in a different application. When the system is designed, it is necessary to determine how many record slots to have active at one time. There should also be safeguards to make certain good data is not over-written.

There are other techniques to determining relative record numbers. Hashing is one technique found in larger systems. The data key is "hashed" using a mathematical formula which provides a location in the file to read the desired record. The problem with hashing is that quite often, two different and distinct data keys will hash to the same numeric value. This situation must be considered when the file mechanism is created.

Another method found in numerically based systems is to simply use the last two or three digits of the number as the relative record number. This is similar to those fast food restaurants that give you the bottom stub of your order to serve as a claim check. That stub usually has a five or six digit number and all but the last two digits are printed in a shaded area. When your order is ready, the clerk calls out the last two digits of the number.

These techniques work well with cyclical data. But what about a scheme where the data tends to be more stable. What about a scheme for indexing names? That will be covered in the next (and last) article.

The Lending Library

Lending activity has picked up now that the word is out that the club lending library has several items for loan. A partial list is included on the next page.

If you are interested, please call 'Gene at 635-8162 before the club meeting to have what you want brought to the meeting. He is also willing to have items picked up at his home between meetings. Please call ahead.

We urge that you loan items you are not using to the library for sharing with other members. Items may be delivered to 'Gene at any club meeting or to his home by appointment. Your items may be returned whenever you like.

BOOKS

"The Last Word on the TI99/4A" by Linda M. and Allen R. Schreiber
(Loaned by Bonnie Snyder)

"Entertainment Games in TI BASIC and Extended BASIC" by Khoo Ton and
Quyen Ton; included cassette tape (Loaned by Kevin Kapus)

"TI Forth Instruction Manual"

"Financial Analysis on TI Computers" by Joseph and Susan Berk,
Chilton Book Co.

"Technical Manual for console, color monitor and interface

RS232 ROM source code listing and schematic

"The SOURCE Manual 1984/85"

MODULES

Football

Video Games I

Alpiner

Hunt the Wumpus

Adventure with two adventure tapes (All loaned by Kevin Kapus)

Personal Real Estate (Loaned by Bill Stites)

Beginning Grammar

Addition and Subtraction 2 (Loaned by Gene Fleming)

CASSETTE SOFTWARE

"Graphing Package" (Loaned by Joe Nuvolini)

NEWSLETTERS

LA 99ers Computer Group, Vol. 3, No. 9

Many others not listed in detail

MAGAZINES

"MICROpendium", Aug. 85

"Home Computer" 22 issues

"Enthusiast 99" 7 issues

WANT ADS

For Sale: One 32K Memory Expansion Card, \$70.00. Contact Sid Win at
596-9334.

If you have any items that you would like to advertise here please contact
the editor before the first of the month.

Copyright 1985

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, only \$3.99 each plus \$1.58 per order for PPM. Entertainment, education, programmer's utilities. Descriptive catalog \$1.99, deductible from your first order.

Tips from The Tigercub, a full disk containing the complete contents of this newsletter Nos. 1 through 14, 50 original programs and files, just \$15 postpaid.

Tips from The Tigercub Vol. 2, another diskfull, complete contents of Nos. 15 through 24, over 60 files and programs, also just \$15 postpaid. Or, both for \$27 postpaid.

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. All for just \$19.95 postpaid.

Nuts & Bolts No. 2, another full disk of 100 utility subprograms in merge format, all new and fully compatible with the last, and with 10 pages of documentation and examples. Also \$19.95 postpaid, or both Nuts Bolts disks for \$37 postpaid.

Tigercub Full Disk Collections, just \$12 postpaid! Each of these contains either 5 or 6 of my regular \$3 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 - my own programs on these disks

are greatly discounted from their usual price, and the public domain is a FREE bonus!

TIGERCUB'S BEST
PROGRAMMING TUTOR
PROGRAMMER'S UTILITIES
BRAIN GAMES
BRAIN TEASERS
BRAIN BUSTERS!
MANEUVERING GAMES
ACTION GAMES
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

For descriptions of these send a dollar for my catalog!

The offer made last month is still good until 1 January - a 10% rebate directly to the user group if one of their members mentions the user group when ordering from me. So far, I've had only 10 responses - and I suspect that 8 or 9 of those didn't even know about the offer!

I goofed again. In the I/O ERROR routine in Tips #28, the ON ERROR STOP will do no good in the place where I put it. It should be placed after the file is opened in line 100 so that it will become the current error trap if the file is opened correctly.

And the CALL KEY example in Tips #28 will look better if R=14. A couple of very knowledgeable programmers have written to tell me that I was wrong, and the manual is right, about CALL KEY status -1. They say that -1 simply means that the same key is being pressed as was pressed during the last keyscan, and that it could have been released and repressed in the interim. This may be, but try this routine and see if you can release and repress a key without getting a status code 0 (no key pressed) and status code 1 (different key pressed) before another status code -1.

```
100 CALL KEY(S,K,S):: PRINT
K,S :: GOTO 100
```

George Steffen has responded to the challenge in the last

Tips, by publishing in the LA 99ers Topics a remarkably compact routine to translate the internal format string representation of numeric data back into numbers. The following lines will update the Menu Loader accordingly.

```
100 !by A. Kludge/M. Gordon/
T. Boisseau/J. Peterson/6. S
teffen/etc.Version 08, 11/85
140 @,00,A,A$,B,C,D$,E,F,FLA
6,1,J,K,KD,KK,M,M$,N$,NM,P,P
$,P6$(1),PP,PP$,Q$,S,ST,T$(1),
TT,VT,V(1),W$,X,X$,Y,K2,S2
810 F=1 :: E=ASC(SE6$(M$,1,1
)):: M=ASC(SE6$(M$,2,1)):: I
F E=0 AND M=0 THEN GOTO 817
ELSE IF E>128 AND M>128 THEN
F=-1 :: E=255-E :: M=256-M
815 FOR I=1 TO 6 :: M=M+(ASC
(SE6$(M$,1+2,1)))/100^I :: M
EXT I :: M=M*F/100^(E-64)
817 PRINT @PP:M
870 FOR P=1 TO NM-1 :: PRINT
@2:P6$(P):TAB(15);V(P,3);TA
B(20);T$(ABS(V(P,1)))::TAB(25
);V(P,2);TAB(31);CHR$(89*ABS
(V(P,1)<0)):: NEXT P :: CLOS
E #2
```

The change in the last line is my own, because it was pointed out to me that the catalog output to the printer did not indicate protected files.

That last line is a good example of the power of relational expressions to accomplish compact programming. The variable V(P,1) picks up its value from the variable A which is read from the disk directory in line 358. This is a number from 1 to 5, indicating the type of file, and if the file is write-protected the number is negative. A true expression has a relational value of -1. If the file is protected, V(P,1)<0 is true, and its value is -1, converted by ABS to +1 and multiplied by 89 to give ASCII 89, converted by CHR\$ to "Y". If not protected, V(P,1) is a positive number, V(P,1)<0 is false and has a relational value of 0; 89 times 0 is still 0, and CHR\$(0) prints nothing.

George also mentioned in a letter that my remarks on the UPDATE mode applied only to VARIABLE files; that RESTORE without a number, to return the record pointer to the beginning of a file, works only with VARIABLE files; that RESTORE with a number works only with

RELATIVE files; and that therefore the only way to RESTORE a SEQUENTIAL FIXED file is to close it and reopen it.

On trying this out, I find that you can write to a FIXED SEQUENTIAL file and still be able to read the following records - but you can't simply "read a record, change it in some way, and then write the altered record back out on the file", as the Reference Guide indicates, because you will change the record FOLLOWING the one you read! It is possible to UPDATE a FIXED SEQUENTIAL file without reading it all into an array and writing it back out, but you must read sequentially to the record you want, close the file, reopen the file, read back to the record just before the one you want to update, then write in the updated record.

I have received several other suggestions regarding the Menu Loader, too many to describe here. You can all modify it to your own tastes and needs. Remember to turn off the pre-scan and ON ERROR while you're working on it, then add any new variable names or CALLs to the pre-scan. And remember, that last line MUST be the LAST line of the program! You can resequence it higher, and change the GOTO accordingly, but don't put anything after it!

I did change my version to slash the zero, since this will carry over into a program that is loaded. If you do this, be sure to add a CALL CHAR to the list in line 150!

```
190 CALL CLEAR :: FOR S=1 TO
14 :: CALL COLOR(S,7,16)::
NEXT S :: CALL COLOR(0,2,16)
:: CALL CHAR(48,"003A444C546
444B0")
```

When you just want to load a program, waiting for it to be read from the disk directory can be a drag. And, you may have trouble recognizing the filename. So, here is the Tigercub Quickloader which I have placed on all my Collection Disks.

First you will need Catwriter, another program that writes a program. This

one will read the disk directory, ignore everything other than programs, ask you for a complete program name for each filename, and write all that into a MERGE format program called CATMERGE.

```
100 !CATWRITER by Jim Peters
on
110 OPEN #1:"DSK1.",INPUT ,R
ELATIVE,INTERNAL :: INPUT #1
:M$,A,J,K :: OPEN #2:"DSK1.C
ATMERGE",VARIABLE 163 :: LN=
1000 :: FN=1100
120 X=X+1 :: INPUT #1:P$,A,J
,B :: IF LEN(P$)=0 THEN 160
:: IF ABS(A)=5 OR ABS(A)=4 A
ND B=254 THEN 130 ELSE X=X-1
:: GOTO 120
130 DISPLAY AT(12,1)ERASE AL
L:P$:" PROGRAM NAME?" ::
ACCEPT AT(14,1)SIZE(25):F$
140 PRINT #2:CHR$(INT(FN/256
))&CHR$(FN-256&INT(FN/256))&
CHR$(147)&CHR$(200)&CHR$(LEN
(F$))&F$&CHR$(8) :: FN=FN+1
150 M$=M$&CHR$(200)&CHR$(LEN
(P$))&P$&CHR$(179) :: IF X<11
THEN 120
160 IF M$="" THEN 180
170 PRINT #2:CHR$(INT(LN/256
))&CHR$(LN-256&INT(LN/256))&
CHR$(147)&SE$$(M$,1,LEN(M$)-
1)&CHR$(8) :: LN=LN+1 :: M$=""
:: X=0 :: IF LEN(P$)<>0 TH
EN 120
180 PRINT #2:CHR$(INT(LN/256
))&CHR$(LN-256&INT(LN/256))&
CHR$(147)&CHR$(200)&CHR$(3)&
"END"&CHR$(8)
190 PRINT #2:CHR$(255)&CHR$(
255) :: CLOSE #1 :: CLOSE #2
```

Next, key in the Quickloader. Do not change the line numbers, do not RESequence, because CATMERGE will be merged into the middle of it and that last line must be the last. Then, enter MERGE DSK1.CATMERGE and then SAVE DSK1.LOAD.

```
100 CALL CLEAR :: DIM M$(40)
:: CALL CHAR(94,"3C4299A1A19
9423C") :: CALL SCREEN(2) :: F
OR SET=1 TO 14 :: CALL COLOR
(SET,15,1) :: NEXT SET :: DIS
PLAY AT(1,4):"TIGERCUB QUICK
LOADER"
110 X=X+1 :: READ M$(X) :: IF
M$(X)<>"END" THEN 110
115 CALL PEEK(8190,A) :: IF A
<>170 THEN CALL INIT
120 R=3 :: FOR J=1 TO X-1 ::
READ X$ :: DISPLAY AT(R,1):
STR$(J);TAB(4);X$ :: R=R+1
:: IF R<23 THEN 150
130 DISPLAY AT(24,1):"CHOICE
? OR 0 TO CONTINUE 0" :: ACC
EPT AT(24,26)VALIDATE(DIGIT)
SIZE(-2):N
140 IF N<>0 THEN 155 :: R=3
```

```
150 NEXT J :: DISPLAY AT(24,
1):"CHOICE?" :: ACCEPT AT(24
,9)VALIDATE(DIGIT):N
160 IF SE$$(M$(N),LEN(M$(N))
,1)="*" THEN DISPLAY AT(12,1
)ERASE ALL:"Return to BASIC"
:: "Type OLD DSK1."&M$(N) ::
STOP
170 CALL CHARSET :: CALL CLE
AR :: CALL SCREEN(0) :: CALL
PEEK(-31952,A,B) :: CALL PEEK
(A&256+B-65534,A,B) :: C=A&25
6+B-65534 :: A$="DSK1."&M$(N
) :: CALL LOAD(C,LEN(A$))
180 FOR J=1 TO LEN(A$) :: CAL
L:LOAD(C+J,ASC(SE$$(A$,J,1))
) :: NEXT J :: CALL LOAD(C+J
,0) :: GOTO 30000
30000 RUN "DSK1.123456789"
```

If you don't want to give your Basic-only programs a filename ending in an asterisk, you can leave out that warning routine, or you can modify it to warn of E/A or MiniMemory programs. If Catwriter has picked up any unloadable program-format files, etc., just delete them from the DATA lines.

The first issue of the GENIAL TRAVELER has arrived, and it is SUPERB! This is a magazine-on-a-disk, a SS/SD floppy loaded with 700 sectors of space of the finest articles and programs you'll ever see! And the programs are ready to run, you don't have to key anything in. The subscription price, until the end of 1985 at least, is \$30 for 6 issues, which computes out to \$5 per disk - many of you are paying your own user group that much for a one-sided disk of public domain!

If the subscribers will only have the guts to refuse to let their friends copy this for free, this venture will surely survive and contribute greatly to the advancement of the TI. The address is - GENIAL COMPUTERWARE, B35 Green Valley Drive, Philadelphia PA 19120.

Gene Burchfield asked if I had a program to print banners vertically. I had never heard of such a thing, so I wrote one.

```
100 DISPLAY AT(12,1)ERASE AL
L:"TIGERCUB STREAMER PRINTER
" !by Jim Peterson
110 DATA 0000,0001,0010,0011
,0100,0101,0110,0111,1000,10
01,1010,1011,1100,1101,1110,
```

```
1111
120 RESTORE 110 :: DIM B$(16
) :: FOR J=1 TO 16 :: READ B$
(J) :: NEXT J :: P$(0)=" " ::
P$(1)=CHR$(230)
130 INPUT "TEXT TO BE PRINTE
D?":T$ :: PRINT :: INPUT "P
RINTER DESIGNATION?":PD$ ::
OPEN #1:PD$
140 PRINT :: INPUT "SIZE? (1
-10)":Z :: IF Z<1 OR Z>10 T
HEN 140
150 FOR J=1 TO LEN(T$) :: A=A
SC(SE$$(T$,J,1)) :: IF A=32 T
HEN GOTO 200
160 CALL CHARPAT(A,M$) :: FOR
M=1 TO 15 STEP 2 :: K$=SE$$(
M$,M,2) :: FOR L=1 TO 2 :: L
$=SE$$(K$,L,1) :: B=POS("0123
456789ABCDEF",L$)
170 M$=B$(B) :: FOR M=1 TO 4
:: N=VAL(SE$$(M$,M,1)) :: M$=
M$&RPT$(P$(N),Z) :: NEXT M
180 NEXT L :: FOR B=1 TO Z/2
+.5 :: PRINT #1:TAB((B1-Z*0
)/2+.5);M$ :: NEXT B :: M$=""
:: NEXT W :: FOR R=1 TO Z/2
+.5 :: PRINT #1:"" :: NEXT R
190 NEXT J :: STOP
200 FOR T=1 TO Z*4 :: PRINT
#1:"" :: NEXT T :: GOTO 190
210 CALL KEY(0,K,S) :: IF S=0
THEN 210 ELSE RETURN
```

If your printer doesn't have the special characters of the Gemini, substitute 00 instead of 230 in line 120, to print X's, or whatever else you want. If you do have the special characters, try some others, such as 239, for this and other graphics printing programs. This routine will print a handy reference chart of them.

```
100 IMAGE 000 0 000 0 00
0 0 000 0 000 0 000 0
110 P$=RPT$(CHR$(251)&CHR$(2
53),21) :: X=0
120 OPEN #1:"PIO" :: PRINT #
1:CHR$(27);"E"
130 PRINT #1:P$:" ASCII COD
ES FOR GEMINI SPECIAL CHARAC
TERS":P$
140 FOR J=160 TO 175 :: K=J-
X
150 PRINT #1,USING 100;K,CHR
$(J),K+16,CHR$(J+16),K+32,CH
R$(J+32),K+48,CHR$(J+48),K+6
4,CHR$(J+64),K+80,CHR$(J+80)
:: NEXT J
160 IF FLAG=1 THEN STOP ELSE
FLAG=1 :: PRINT #1:"":P$:"
TI-WRITER CODES FOR GEMINI
SPECIAL CHARACTERS":P$ :: X
=128 :: GOTO 140
```

Another one that just looks pretty -
100 !KALEIDOSPRITES by Jim P
eterson
110 CALL CLEAR :: FOR CH=100
TO 128 STEP 4 :: FOR L=1 TO

```
4 :: RANDOMIZE :: X$=SE$$(
0018243C425A667E8199A5BDC3D8
E7FF",INT(16&RND+1)*2-1,2)
120 B$=B$&X$ :: C$=X$&C$ ::
NEXT L :: CALL CHAR(CH,RPT$(
B$&C$,4)) :: B$,C$="" :: NEXT
CH :: Z=2 :: CALL SCREEN(5)
130 CALL MAGNIFY(2) :: K=1
FOR J=1 TO 7 :: S=96+4*J ::
R=16*J :: C=100&RND+20
140 IF J>5 AND Z=4 THEN T=5
:: GOTO 160
150 T=INT(15&RND+2) :: IF T=5
THEN 150
160 CALL SPRITE(0,K,S,T,R,C,0
K+1,S,T,177-R,C,0K+2,S,T,R,2
41-C,0K+3,S,T,177-R,241-C) ::
K=K+4 :: NEXT J
170 Z=INT(2&RND+1)*2 :: GOTO
130
```

100 !DISK MATCHER by Jim Pet
erson
110 DISPLAY AT(8,9)ERASE AL
L:"DISK MATCHER" :: "To c
ompare a backup disk:"with
a master and list any "file
s found on one but not"
120 DISPLAY AT(15,1):"on the
other." :: " Press
any key"
130 CALL KEY(0,K,S) :: IF S=0
THEN 130
140 DISPLAY AT(12,1)ERASE AL
L:"INSERT MASTER - PRESS ENT
ER" :: CALL KEY(0,K,S) :: IF
S=0 THEN 140

```
150 OPEN #1:"DSK1.",INPUT ,R
ELATIVE,INTERNAL :: INPUT #1
:D$,A,J,K :: DIM F$(127)
160 X=X+1 :: INPUT #1:F$(X)
,A,J,B :: IF LEN(F$(X))<>0
THEN 160 ELSE CLOSE #1
170 DISPLAY AT(12,1)ERASE AL
L:"INSERT BACKUP DISK": "PR
ESS ENTER" :: CALL KEY(0,K,S
) :: IF S=0 THEN 170
180 OPEN #1:"DSK1.",INPUT ,R
ELATIVE,INTERNAL :: INPUT #1
:D2$,A,J,K :: DIM F2$(127)
190 Y=Y+1 :: INPUT #1:F2$(Y)
,A,J,B :: IF LEN(F2$(Y))<>0
THEN 190 ELSE CLOSE #1
200 DIM F(127) :: FOR J=1 TO
X :: FOR L=1 TO Y :: IF F2$(
L)=F$(J) THEN F(L)=1 :: GOTO
220
210 NEXT L :: PRINT F$(J);"
NOT ON BACKUP"
220 NEXT J
230 FOR M=1 TO Y :: IF F(M)=
0 THEN PRINT F2$(M);" NOT ON
MASTER"
240 NEXT M :: END
A very useful tip from Jim
Swedlow, in the Orange
County ROM newsletter -
INPUT respects any trailing
print separator on a
preceding PRINT command. Try
it -
100 PRINT TAB(20);: INPUT B
$
MEMORY FULL IN LINE 400
Jim Peterson
```

First a few announcements ... This month we are beginning a new program. When a new disk fills up at the library, we will offer it as a special. The first will be available at the January meeting and will include all of the programs mentioned in this review. Now before we review the individual programs I want to mention a few updates to programs we already have. First, Gothic, #395, has been updated by member Rich Houghland. He added some new things to the program. You can now select normal, condensed, or double sized. Version 2.2 of Disk Manager has arrived and with it some news about the program. The program is a FREeware program, the residuals of which go to the Ottawa User's Group. Therefore, if you have the program and haven't already sent the group some green, please do so as it is an excellent program and they deserve to be compensated. The two files replace MGR1 & MGR2, #386/7. The new version corrects some of the bugs in the previous version and also allows you to read DV80 files as option 3 under the Files area. Also, there is a new TI-Writer loader that returns the SD function. This one is a must for all serious TI-Writers. Now to the new program additions. Programs #446/7 are 40-Column and Example of 40-Column(8). They are an XB programming aid that allows you to get 40 columns while programming in XB. More Bach(5), #468, is more music by Bach for you music lovers. The XB FORTH Loader(8), #469, is an XB utility that allows you to load a forth disk from XB, a handy routine. Important Numbers(5), #470, is an XB utility that keeps track of either serial numbered items or credit cards. You can enter the item, ser. no., model, and purchase date. On the credit card side you can enter credit card name, number, and notes about it. Program #471 is 10-day Journal(7). It is an XB program that will keep track of your appointments for a ten day period logging them off to disk. Programs 472-4 are List 132, List 28 and LISTDOC(7), the accompanying documentation for the former two. These are assembly routines that are called from XB and will allow you to list programs in either 28 columns as they are on the screen, or in 132 columns in condensed print, which will save you some space. Disk Indexer(9), #475, is an excellent XB utility. I put it on the special disk as LOAD and it automatically catalogs the disk and then allows you to delete files, run programs both XB and Assembly. It will allow you to input comments about each program and then print out a listing of them. It is an outstanding program! Make a Loader(7), #476, is a similar program but not as sophisticated. When you run this program it asks you which files on your disk you want in a loader. It then saves off in a merged format a loader program that you can MERGE into an empty console and then resave it as a program. If you save it as LOAD when you select XB it will list the programs you have selected and allow you to auto load them. TV-Alignment(7), #477, is a neat little program that will display colorbars, gray scale, cross hatch, red raster, and green raster so you can align your TV/Monitor. Program #478 is Warning Labels(7). It is a utility that I reworked a bit. It now prints two-across warning labels you can put on disk mailers that reminds the postal service folks not to bend or x-ray the mailer. Finally, here is another great utility called XB Loader(9), #479. It is an XB program that will allow you to load an assembly program in lieu of using Option 5 of the E/A. It is a freeware program and although short, I guess the author deserves something for figuring out how to do it so nicely, but not an awful lot! The final program this month is #480, Disk Envelope(7). It is an XB utility, written by Bonnie Snyder, that will enable you to make a disk envelope out of an 8x10 sheet of paper and a printer. That's it for this month. This first special of 1986 is a really great one! See y'all next time!

No.	Name	Type	Lng	Date	File Name	Dsk
466	40-COLUMN	PROGMG	XB	NOV 85	40-COL	33
467	EXMPL OF 40 COL	PROGMG	XB	NOV 85	40-EXMPLE	33
468	MORE OF BACH	MUSIC	XB	N/A	BACH2	33
469	XB FORTH LOADER	UTIL	XB	N/A	EXBFORTH	33
470	IMP'NT NUMBERS	UTIL	XB	N/A	IMFTNUMS	34
471	10-DAY JOURNAL	UTIL	XB	N/A	JOURNAL	34
472	LIST 132	UTIL	ASY	N/A	LIST132	33
473	LIST 28	UTIL	ASY	N/A	LIST28	33
474	LIST DOCMNTN	UTIL	VB	N/A	LISTDOC	33
475	DISK INDEXER	UTIL	XB	N/A	LOADX	34
476	MAKE A LOADER	UTIL	XB	N/A	MAKELOADR	34
477	TV ALIGNMENT	UTIL	XB	N/A	TV/ALIGN	34
478	WARNING LABEL	UTIL	XB	N/A	WARN/LAB	33
479	XB LOADER	UTIL	XB	N/A	XBLOADER	34
480	DISK ENVELOPE	UTIL	XB	N/A	D/ENV	34

This publication is printed monthly for the benefit of the membership of the Front Range 99'er Computer Club. The Club and the newsletter is not for the benefit nor is backed by any commercial enterprise. Both are non-profit in nature and are for the sole purpose of computer education. Any fees collected are used to defray any cost to maintain the club and publish this newsletter. Neither the newsletter nor the Club have any affiliation with Texas Instruments. Any statements in this newsletter are not necessarily the opinion of the membership.

We do participate in an exchange program with other clubs. Any article printed in this newsletter can be copied or reproduced with proper credit to the newsletter and it's author.

We do offer advertising space to businesses interested in doing so. Our rates are as follows:

1/4 page.....	\$ 10.00/month
1/2 page.....	15.00/month
full page insert.....	25.00/month
full page [we copy].....	30.00/month

We offer out of state subscriptions to the newsletter at a rate of \$10.00 per year.

Club memberships are \$15.00 for individual and \$21.00 for family, per year. This includes a subscription to the newsletter and other club benefits.

FRONT RANGE 99'er COMPUTER CLUB
P.O. BOX 9572
COLORADO SPRINGS, CO 80932



FIRST CLASS POSTAGE

86/07
BRYCE WILCOX
15425 CLEARWATER DR
ELBERT CO 80106

#144