NEW HAMPSHIRE 99'ERS

DECEMBER 1984

NEWSLETTER

NHUG NEWSLETTER - PO BOX 7199 HEIGHTS STATION - CONCORD, NH 03301

(((((SEASON'S GREETINGS)))))

This month I will attempt to keep this column short! Here goes...

First and foremost, because of the holiday season, there will be NO December meeting! The January meeting has been scheduled for January 23, 1985. Please mark your calendars.

Our library is desperately in need of an "overhaul". Since all the programs are currently on disks, we need members with expanded systems to volunteer to run, evaluate, debug, comment on etc. programs. If you would like to help, please call or write: H M LaBonville - 121 Camelot - Bedford, 03102 - 603/472-3369.

Glenn Hammond recently experienced problems with his console and found the Exchange Center to be out of them. He had to wait three weeks for shipment. Please, to save time and money, isolate the problem in your equipment by substituting equipment with a friend or at a NHUG meeting; call the Exchange Center to see if they have a replacement and if so, the cost; when shipping, package ONLY the defective component(s) and pack properly to avoid postal damage and enclose a brief note describing the equipment failure. The nearest center is: TI CUSTOMER SERVICE CENTER #811 - 504 Totten Pond Road - Waltham, MA 02154 (617/895-9161). See October NHUG NEWSLETTER for some exchange costs.

(((((BULLETIN BOARD)))))

INDECO - 133 W Chapman Ave, Suite A - Fullerton, CA 92632 (800/321-1721). Their catalog has a limited number of TI related goods, but their educational programs look quite interesting. Requirements are console, recorder, and in some cases speech. They claim that "anything you order from INDECO is unconditionally guaranteed to please you... If you don't like it, return it within 30 days and your purchase price will be refunded promptly and cheerfully". If you have questions, call them, they want to hear from you. Then call us, we'd all like to know if they're serious to their commitment.

CAHILL ELECTRONICS - 91 Main St - Kingston, NH 03848 (603/642-8469). They sent a flyer announcing the availability of 99/4 (A?) KEYBOARDS. 48 keys - 4" X 10" Black or Gray. $7.50 each postpaid.

ZYLA'S - Merrimack, NH 03054. We noted that they had rebuilt GE Computer Recorders for $20 each.

UNISOURCE - PO Box 64240 - Lubbock, TX 79464 (order 1-800/858-4580). They forwarded us their latest Encyclopedia Catalog. It is an excellent reference of 99/4A software and hardware. The current price list includes specials on the Exceltec (Sunware) NEW EXTENDED BASIC ($69.95) and Epson RX 80 printer ($264.95).

ONLINE (1-800-438-2438). ONLINE is a new BBS type of system. It is a free service that lists ads for computer related products from companies, computer Newsletters and for a small fee you may place ads in it. Be sure to sign up you get their free monthly Newsletter in the mail. Current hours of operation:

Monday thru Thursday: 5 pm to 1 am CDT
Friday thru Monday: 5 pm to 5 am CDT

LECHMERE - Mall of New Hamp. - So Willow Street - Manchester. They sell single diskettes for 99 cents each. Do check for dings and make sure they have hub rings. They WILL exchange "bad" disks. We find that we must return ten percent of our bulk purchases.

STAR MICRONICS, INC. - 3 Oldfield - Irvine, CA 92714 (attn: Cherie Maddocks, Tech Support). Star Micronics has published an addendum to their Gemini Users Manual for the 99/4A. This free addendum contains a 12 page list of special instructions and programs.

HOW TO FIX A DISK

Niraj N. Shah - Mike Ballmann

This article originally appeared in the Spirit of 99, The Newsletter Bulletin Board of the Central Ohio Ninety Niner's. The Board # is 614/451-0880.

Did you ever try to catalog a disk only to find out the Disk Controller thinks the disk is NOT initialized? But you know better! What do you usually do with the blown disk? Most people Delete the file giving them the Problem. Usually that does correct the Problem, but it also gets rid of that file forever. The ultimate solution is to use DISK FIXER by Navorone Industries.

The DISK FIXER enables one to examine and change the contents of any disk on a sector-by-sector basis. I think it is well worth its forty-dollar list Price. It is available from TI retailers or directly from Navarone. Here is the Process to fix a blown disk using DISK FIXER.

Get a hardcopy catalog of the blown disk, or even better, get a complete (old) catalog of what should be on the disk. If a complete catalog is unavailable try to remember what should be on the disk and write those names down on Paper. Once you have a catalog, you are ready to start using DISK FIXER.

Insert the DISK FIXER cartridge and select option "2" from the TITLE SCREEN. You should now see the DISK FIXER MENU. Do the following if the most recent catalog of the bad disk tells you there are more sectors used/free than is logically possible: 358 for single-sided; 718 for double-sided disks. For example, IF the catalog lists 500 sectors used/free on a single-sided disk THEN do the following ELSE GOTO the paragraph on "SECTOR ONE".

This part tells how to fix up Sector 0; which is the sector containing the information concerning the disk name and number of sectors used/free on the disk. If the disk catalog tells you the used/free sector is in error then Sector 0 needs to be fixed. The easiest way to do this is to copy a good sector 0 from another disk to the blown disk:

1) Insert a good disk in your drive.
2) Read Sector 0 of that disk: R 0,1 [ENTER]
3) Now Put the blown disk in the drive
4) Write good Sector 0 to disk: W 0,1 [ENTER]

If you catalog the bad disk, you will see that the diskname and not the used/free info is the same as the good disk, but do not let that alarm you. We did that to fool the Disk Controller into thinking the bad disk is at least Partially restored to normalcy. Now we need to fix up the blown disk as much as we can. This is done by changing Sector 1.

Here's how to fix Sector 1: First, get the most complete catalog and and the most recent catalog of the bad disk in front of you. Then compare the two to see which filenames are missing. Next, compile an alPhabetical list of all the filenames which are and should be in the catalog. Then you need to find the corresponding sector for each filename of the DISK FIXER.

1) Put the bad disk in the drive.
2) Find a filename by: F 0,2D0,1 [ENTER] then [ENTER] the filename.
3) Ignore the "ERROR IN SECTOR" message.
4) Write down the sector number for that filename.

5) If that filename could not be found make sure you typed it in correctly and try again; otherwise that file does not exist on that disk.
6) Repeat the process from step two for all of the filenames.

You should now have an alphabetic list consisting of two columns: filenames and sectors. With that info in hand you are ready to begin fixing up the bad disk. This is done by modifying Sector 1 of the blown disk. First you have to read Sector 1 from the bad disk by doing this:

1) Put the bad disk in the drive.
2) Read Sector 1 of the disk by: R 1,1 [ENTER]

Then you want to alter the contents of Sector 1. This is done by using the alter function of DISK FIXER. This process is best learned by observing a concrete example. Let's say the blown disk has 14 files (filenames) on it. The rest of the sector should be all zeros. Let's alter Sector 1:

1) Keep the bad disk in the drive.
2) Enter the Alter function: A 0 [ENTER]
3) Type the following just as shown, including the spaces: 1 2 3 4 5 6 7 8 9 A B C D E
4) DON'T press [ENTER] yet!
5) If you saw a non-zero entry after the E entry in the first column then type in [0] and a [SPACE] and repeat until the first colum shows a zero.
6) Now press [ENTER].
7) Write the revised Sector 1 to the bad disk: W 1,1 [ENTER]

You have just entered a table of pointers to the files on the disk. The table points to the corresponding sector for each filename. This is the table that is updated and sorted if you add/delete files to the disk.

Leave the DISK FIXER by typing [Q] for QUIT and press [ENTER]. Then catalog the disk. Let's call this new catalog the mixed catalog. You will see the reason once the disk has been cataloged. Notice how the catalog is NOT in alphabetical order. It does, however, contain all of the filenames that you hoped and prayed would be on the disk! The next step is to alphabetize the catalog. This is done by first alphabetizing the catalog on paper and carrying along the appropriate sector number of each filename. Here is an example of a Mixed Catalog:

MIXED FILENAME	CATALOG SECTOR	SORTED FILENAME	CATALOG SECTOR
CAT	1	APPLE	E
SCREEN	5	CAT	1
VOTE	2	DEMO	7
FIRE	6	FIRE	6
APPLE	E	HELLO	9
HELLO	9	JUSTIFY	D
SCROLL	C	LOAD	3
LOAD	3	LOGO	A
TIME	8	PLOT	B
DEMO	7	QUICK	4
QUICK	4	SCREEN	5
JUSTIFY	D	SCROLL	C
PLOT	B	TIME	8
LOGO	A	VOTE	2

The above example shows how you should alphabetize the filenames and the corresponding sector numbers on paper. If your are unsure when dealing with funny characters, the system alphabetizes by lower ASCII values. These values can be found on your TI BASIC reference card. Once you have done this you are ready to enter this info into Sector 1. You do not have to enter the filenames, just the sector numbers:

1) Put the blown disk in the drive.
2) Read Sector 1 by entering: R 1,1 [ENTER].
3) Enter the Alter function: A 0 [ENTER]
4) Type in the sector numbers in order. In the above example, that means: E 1 7 6 9 D 3 A B 4 5 C 8 2 [ENTER].
5) Write the revised sector to disk: W 1,1 [ENTER]
6) Put a Write-Protect tab on the disk!

You have now fixed the disk. For verification, quit the DISK FIXER program and catalog the disk. You should have no problems during the cataloging process. But you are not completely done yet! DO NOT add/delete any files or programs to this disk!

Get a fresh disk and initialize it to the same configuration as the blown disk. Then back-up the bad disk to the fresh disk. Then catalog the fresh disk and you will see that the used/free sector info is now correct. Thus, the fresh disk is now your working disk and the blown disk is now a disk for your archives just in case you remember a file that was not previously recovered from it. Go through the above procedures to recover that new-but-old file.

If you have any questions on this article, Please leave private mail to MIKE BALLMANN... Happy Fixing!!

TI TIPS !!!

Here are a few simple "PEEKS POKES" that can be helpful with your TI home computer.

** Some times you may find it beneficial to disable some of the functions. Remember to use the command CALL INIT in your program so from now on I won't mention that any more.

To disable the following functions type CALL LOAD(-31806,X) make x= the number indicated for the specific function below:

If X=16 disables the FCTN QUIT
X=32 disables the sound processing
X=48 disables the Quit Key and Sound
X=64 disables Sprite motion
X=80 disables Quit Key and Sprite motion
X=96 disables Sprite motion Sound
X=128 disables Spite motion, Sound Quit

** To unprotect a program that has been protected with extended basic type CALL LOAD(-31931,0) Then the program will list and save in the usual manor.

** For people with a 32K memory expansion you can free up around 8K of memory with CALL LOAD(-31866,33,0) Type SIZE before and after. The only thing is I'm not sure that you will be able to access this extra memory but it may get you thinking?

** Because the disk drive uses so much memory (about 2k) you may not be able to load some programs from cassette. You can turn off your drive and reinitialize or;

Type CALL LOAD(-31888,63) to make the drive disappear.
CALL LOAD(-31888,55) to make the drive reappear.

** As most people I use to end my program with the command STOP or END. But some times it is better to end up at the title screen instead of in X-Basic with the old program in memory. Just replace the END statement with; CALL PEEK(2,A,B) :: CALL LOAD(-31804,A,B)

NOTES ON THE T.I. ADVANCED ASSEMBLY LANGUAGE DEBUGGER

Richard J. Bailey
68A Church Street
Gonic,N. H. 03867

One good buy from the user group is the debugger disk. Unfortunately, the description of the disk is incorrect. The disk can be used not only with the EDITOR/ASSEMBLER module but also with T.I. basic and extended basic. You still have to have 32K added memory and the disk drive to use the SUPER-DEBUGGER. The HELP file on the disk can be printed or viewed on the screen with one of the following short programs. It isn't too practical to view the HELP file on the screen since it is 7 printed pages long! If you use the printer version, change the printer name in line 120 to agree with your printer and use drive 1 if you have more than one drive. The HELP file is basically an instruction manual on disk. With the screen version you can press and hold 'ENTER' to halt the listing.

```
100 CALL CLEAR
110 OPEN #1:"DSK1.HELP"
120 LINPUT #1:B$
130 PRINT B$
140 CALL KEY(0,KY,ST)
150 IF KY=13 THEN 140
   160 GOTO 120
```

```
100 CALL CLEAR
110 OPEN #1:"DSK1.HELP"
120 OPEN #2:"RS232.BA=2400.DA=8"
130 LINPUT #1:B$
140 PRINT #2:B$
150 GOTO 110
```

Enter and run the following program to see how you could enter the debugger with basic or extended basic. Refer to the help file for more instructions on its use.

```
100 CALL CLEAR
110 CALL INIT
120 INPUT "PUT DEBUGGER DISK IN        DRIVE 1,
      PRESS ENTER ":DUMMY$
130 PRINT
140 PRINT "PRESS 'N' WHEN THE LOADING  STOPS.
    ENTER PRINTER NAME   THEN 'DA000 A140'
    TO DUMP   TO THE PRINTER."
150 CALL LOAD("DSK1.SBUG")
160 CALL LINK("SBUG")
```

S U R V I V A L

Those of you who are members of a TI users group or are reading this do to your interest in the 99/4A are hopefully more strongly motivated towards the survival of your machine than is the public at large. You had better be because the general public and owners of most other types of computers have considered us dead and buried for almost a year now. Even befoe the TI pullout, it was an uphill battle to fight the opinion that the 99/4A was more than a not so cheap, and then later, a cheap toy. Peripherals and software were mostly TI controlled and frequently unavailable.
Now, one year later, I still have to defend my machine every time I talk to a non TI owner. Besides trying to convince someone that this is the best machine available(YES, it is still available. You just won't find it in a K-Mart or Sears.) I must also convince people that it is still alive and even kicking pretty hard.
There is more software out now than has ever been available before. There is more hardware available now with quite a bit soon to hit the market. If you are not familiar with the TI hardware and software situation, then contact a TI users group and talk to the members and officers.
I frequently speak to people who decry their lack of information about TI developments. These same people are surprised by my descriptions of the new software and hardware that has come out.
This article is about SURVIVAL. It is a plea to any and all 99/4A supporters out there. Production for this machine may stop, or just maybe continue on indefinitely. Rumors still abound! It will definitely stop unless there is more support from the 99/4A buying public.
Two excellent sources of information nowadays are MICROpendium, P.O.Box 1343, Round Rock, TX 78680 *$12 yearly subscrition* and The Smart Programmer by Miller's Graphics, 1475 W. Cypress Ave., San Dimas, CA 91773 *yearly subscription $12.50*.
I put MICROpendium first because in it's less than a year's existence it has been produced consistently and remained high quality. MICROpendium is newspaper similar with news, reviews, feature articles, personal interviews, Users' Group tidbits, and advertisements.
The Smart Programmer is styled like a newsletter. It does not have advertisements but is instead aimed at those wanting to improve their programming and technical knowledge. A non programmer would still find this very worthwhile.
The jury is still out on what used to be the best support for the TI 99/4A, the Home Computer Magazine (formerly 99'er Magazine). In the most recent issue (available since Oct.84), it promises in the beginning pages to correct all its past mistakes. A number of faults that a magazine may have are listed. Those notable faults are: reviews biased in favor of advertisers and an inconsistent and unreliable publishing schedule. Gary Kaplan, publisher, does not actually come out and say, "Hey folks! The 99'er Home Computer Magazine has been guilty of this!". What he does say is as close to this as we could ever expect to get. The most recent issue is a distinct improvement over the previous three. Those three issues covered from December 83 to September 84. Three issues in ten months for a 'monthly' magazine.
Speaking of juries being out let's turn to the business that is called the 'International 99/4A Computer Users Group' run by Charles LaFara. I disagree with a number of their policies and the status of whether they will live or die remains to be seen. Even after making those statements, I must admit they were a viable force for bringing together programs, groups, and users from around the country.
Reservations exist as to whether the Home Computer Magazine will exist three months from now, much less a year from now. Despite those rreservations, I buy the magazine and inform others about it. I used to do the same for Charles LaFara's business.
At the present time we have two good sorces: MICROpendium and The Smart Programmer. We also have two questionable sources: Home Computer Magazine and 99/4A IUG. If what we have available to us isn't supported by the 99/4A public at large, we will soon have nothing.
Going beyond publications, we have other avenues of support available to us. One of those is Guy-Stefan Romano's HELPLINE -(415)753-5581. This is a technical and general information source available at no charge. Mr. Romano and his HELPLINE have been spoken of before in this and other newsletters. I mention it now to remind people of a real and viable service available to them. I also wish to make a suggestion. The Atlanta C.U.G. has included Mr. Romano on our newsletter mailing list. If other groups did the same, Mr. Romano would be kept better informed about TI information and developements around the country. The HELPLINE Guy-Stefan Romano runs is done as a Public Service. He receives no renumeration. His receiving newsletters from the various groups would benefit us all.
Another type of phone support that we have is modem. There are now over a hundred Bulletin Board Systems around the country which are running on the TI 99/4A computer. At another time we will publish a listing of some of them. The numbers are also listed on most of the TI boards. Right now a modem gives the TI owner a wide avenue to the 99/4A community.
Two other services available to those with modems are the SOURCE and COMPUSERVE. Both have special sections where TI related topics and messages are left. To participate in the one on the SOURCE type: POST R TI from Command level. To get to the TI SIG on COMPUSERVE type: GO PCS27.
Don Bynum (SOURCE mail address- TI9998) was in charge of the Home Computer Division at Texas Instruments. Mr. Bynum is working to get the TEXNET feature of the SOURCE back up and running.
We, who own TI computers have some very good resources. I am just worried that those resources will not be supported and will consequently dry up and blow away.
There is one more very vital subject to be covered under the topic of what can we, the users do to support our machine so it will flourish rather than wither. That subject pertains to the copying of purchased software. Many people do it. We all know that. I don't really think that those doing it will stop. I am not in a position to lecture on the matter, but I can state my feelings. I feel that if copying becomes so very prevalent and accepted! it will kill the production of the quantitty and quality programs we have begun to see on the market. I know, just like any realist knows, that copying will continue. I know that it is impossible to get people to turn down a good program that is just given to them from another at no charge. Copying may make you feel guilty but there is a much worse result. It is similar to cooking the chicken that lays the eggs that you eat. You may get a supper out of it, but you will not get any eggs ever again. Copying may do us in this way.
Is there a solution? If there is, it involves the people doing the copying spending money. Copying usually goes on in cloes knit 'circles' and there are <u>many</u> of these 'circles' around the country. Fact of Life: Those doing the copying <u>will not stop!</u> If the people in one of those 'circles' each contributed money once a month and purchased software, the cost to individuals would be nominal yet software would be bought and paid for each month on a regular basis. The end result would be more money spent on programs and their developers. I know it is fantasy to think of all those many groups around the country who copy programs doing something like this, but I had to put my 2 cents worth in.
Support! Our computers will not <u>survive</u> if we do not support the Users Groups, information sources, 99/4A publications, and the software producers. I am one of those who "never say die", who will hang in there and cling to every ray of hope and rumor until past the point that there is no hope. I can only hope that I am not alone, so this computer will live.

Gary Matth...

TIPS FROM THE TIGERCUB

#16

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.

These Tips are being mailed, together with my new catalog #5, to every Users Group that I know of. I hope that you will make both the Tips and the catalog available to your membership. I am sorry that I cannot take out paid ads in your newsletters, but to advertise in each one of them would cost me more than I have made in the past 6 months, and I would not get enough business to break even.

If you would like to continue receiving these Tips, put me on the mailing list for your newsletter, and give me some indication that my Tips are really reaching your members and not going into someone's private file. If I receive enough business from this mailing to pay for its cost, I will then continue to send you my Tips. If not, this will be the last issue of the Tips from the Tigercub.

Copies of my catalog are available for $1.00, which is deductable from your first order. I have over 130 absolutely original quality programs in Basic, many of them now also available in XBasic, on casette or disk for only $3.00 each plus $1.50 per order for casette, package and postage, or $3.00 for diskette, package and postage (higher overseas). I give one-day service, I give bonuses for repeat orders, I give bonus programs on diskette orders.

In addition, any User's Group member who mentions his/her users'group when sending me an order before 1 Jan. 1985 may deduct 10% from the cost of the programs.

Tips from the Tigercub #1 thru #14 are now available, with more added, as a diskfull of 50 programs, routines and files for only $15 postpaid.

I have also now completed my NUTS & BOLTS disk of 100 XBasic utility subprograms in MERGE format, ready to merge into your own programs, for just $19.95 postpaid.

In The last Tips, I mentioned that I wished I knew who to credit for that remarkable routine to redefine the cursor. Dave Peden has written me that credit should be given to Terry L. Atkinson of 28 Savona Ct., Dartmouth, NS B2W 4R1 CANADA.

And I would like to strongly recommend that you support the 99'ers Users Group Association, 3535 So. H st., #93, Bakersfield CA 93304. They are a strictly non-profit group, devoting a lot of time and effort to helping us all, and they publish a great newsletter..

Every Tips must include a bit of music, and my grandson has requested that I pass this one on to all other two-year olds.

```
100 !ALPHABET SONG - by Jim
Peterson
110 DIM N(21)
120 CALL MAJORSCALE("C",N())
130 CALL SCREEN(5):: DISPLAY
 AT(24,1)ERASE ALL:"READY -
TYPE THE ALPHABET" :: CALL M
AGNIFY(2)
140 CALL KEY(3,K,ST):: IF (S
T<1)+(K<65)+(K>90)THEN 140 :
: CALL SPRITE(#1,K,16,96,120
):: IF K=87 THEN GOSUB 220 E
LSE GOSUB 200
150 IF (K=90)&(FLAG=0)THEN 1
60 ELSE 140
160 FLAG=1 :: M$="C115566D5C
443322D1" :: T=150
165 FOR J=1 TO 18 :: CALL SP
RITE(#J,64+J,INT(11*RND+6),9
6,128,J*5,J*5)
170 X=ASC(SEG$(M$,J,1)):: IF
 X>58 THEN T=150*(X-64):: GO
TO 190
180 X=X-48 :: CALL SOUND(T,N
(X),0)
190 NEXT J :: FLAG=0 :: CALL
 DELSPRITE(ALL):: GOTO 140
200 Y=VAL(SEG$("115566544332
22215543325332",K-64,1))
210 CALL SOUND(500,N(Y),0)::
 RETURN
220 CALL SOUND(500,N(5),0)::
 CALL SOUND(500,N(5),5):: CA
LL SOUND(500,N(4),0):: RETUR
N
230 SUB MAJORSCALE(K$,N())
240 F=VAL(SEG$("110123131147
165175196",POS("ABCDEFG",K$,
1)*3-2,3))
250 C$="10101101010110101101
0101101011010101"
260 FOR J=1 TO 36 :: IF SEG$
(C$,J,1)="0" THEN 280
270 X=X+1 :: N(X)=F*1.059463
094^(J-1)
280 NEXT J :: SUBEND
```

Lines 230-280 of that routine are an example of the kind of handy-dandy subprograms you will find on my Nuts & Bolts disk.

We haven't had a Tigercub Challenge for some time, so -

How can you store a hundred or more values of any size, positive or negative, integer or non-integer, even in exponential notation, without dimensioning an array or opening a file?

Now, how can you link your program to another by a RUN statement, thereby losing all data, and recover those values? Yes, I know you can save them on the screen and read them back, but can you find a better way?

Here's a little demo program of how motion can be created by the repetitive redefinition of characters. I call it ETERNITY.

```
100 CALL CLEAR :: CALL SCREE
N(2):: CALL COLOR(1,16,1)::
CALL CHAR(33,"",34,"",35,"",
36,"")
120 FOR R=1 TO 12 :: CALL HC
```

```
HAR(R,R+4,33,26-R*2):: NEXT
R
150 FOR R=13 TO 24 :: CALL H
CHAR(R,29-R,34,(R-12)*2):: N
EXT R
180 FOR C=5 TO 16 :: CALL VC
HAR(C-4,C,35,34-C*2):: NEXT
C
210 FOR C=17 TO 28 :: CALL V
CHAR(29-C,C,36,C*2-33):: NEX
T C
225 FOR J=0 TO 7 :: A$(J+1),
B$(8-J)=SEG$("00000000000000
",1,2*J)&"FF" :: NEXT J
230 C$(1),D$(8)=RPT$("80",8)
:: C$(2),D$(7)=RPT$("40",8):
: C$(3),D$(6)=RPT$("20",8)::
 C$(4),D$(5)=RPT$("10",8)
240 C$(5),D$(4)=RPT$("08",8)
:: C$(6),D$(3)=RPT$("04",8):
: C$(7),D$(2)=RPT$("02",8)::
 C$(8),D$(1)=RPT$("81",8)
250 FOR C=2 TO 15 :: FOR J=1
 TO 8 :: CALL CHAR(33,A$(J),
34,B$(J),35,C$(J),36,D$(J)):
: NEXT J :: CALL SCREEN(C)::
 NEXT C :: GOTO 250
```

Next, I would like to share with you a gem of a "why didn't I think of that" routine which John Taylor sent me.

```
100 ! 28 COLUMN TEXT ROUTINE
IN EXTENDED BASIC (EASILY
CONVERTED TO BASIC) BY JULIE
PACK, B.U.G., P.O. BOX 1402
PALM BAY, FL 32906
110 ! ENHANCED BY JET
SHOALS SYSTEMS, P.O. BOX 2926
MUSCLE SHOALS, AL 35662
120 CALL CHAR(64,"00282828")
130 ! PROGRAM TO COPY STARTS
      HERE
140 CALL CLEAR :: X=-1
150 RESTORE
160 IF X>=21 THEN X=1 :: CAL
L WAIT
170 READ MESS$
180 IF MESS$="P" THEN DISPLA
Y AT(X+2,1):Z$ :: X=X+4 :: Z
$="" :: GOTO 160
190 IF MESS$="ZZZ" THEN DISP
LAY AT(X+2,1):Z$ :: CALL WAI
T :: END
200 IF LEN(Z$)>0 THEN MESS$=
Z$&" "&MESS$
210 X=X+2
220 IF X>=21 THEN X=1 :: CAL
L WAIT
230 IF LEN(MESS$)<29 THEN DI
SPLAY AT(X,1):MESS$ :: Z$=""
 :: GOTO 160
240 FOR A=1 TO 29
250 I=POS(MESS$," ",A)
260 IF (I=0 OR I>29)AND A=1
THEN A,J=29 :: GOTO 290
270 IF I=0 OR I>29 THEN A=29
 :: GOTO 290
280 J,A=I
290 NEXT A
300 IF X>=21 THEN DISPLAY AT
(X,1):SEG$(MESS$,1,J-1):: X=
-1 :: CALL WAIT :: GOTO 320
310 DISPLAY AT(X,1):SEG$(MES
S$,1,J-1)
320 IF SEG$(MESS$,J,1)=" " T
HEN I=1 ELSE I=0
330 Z$=SEG$(MESS$,J+I,163)::
 MESS$=Z$ :: IF LEN(Z$)>28 T
HEN X=X+2 :: GOTO 240
340 GOTO 160
350 DATA "THIS SHORT ROUTINE
 WILL ENABLE YOU TO WRITE LO
NG TEXT MATERIAL IN YOUR DAT
A STATEMENTS SO YOU WON'T HA
VE TO WORRY ABOUT COUNTING"
360 DATA "THE LENGTH OF YOUR
 SENTENCES ALL THE TIME.  TH
IS ROUTINE WILL AUTOMATICALL
Y EDIT YOUR TEXT TO FIT A 28
 COLUMN SCREEN."
370 DATA "A SUGGESTION- IT I
S A GOOD IDEA TO PUT A QUOTE
 AT THE BEGINNING AND END OF
 THE DATA STATEMENTS SO YOU
WON'T HAVE TO WORRY ABOUT"
380 DATA "COMMAS LIKE THIS ,
,, AND THEY WILL REMAIN IN Y
OUR TEXT PROPERLY."
390 DATA "THIS ROUTINE WILL
ALSO CLEAR THE SCREEN (WHEN
FILLED) AND CONTINUE READING
 YOUR DATA AND DISPLAYING YO
UR TEXT ON THE NEXT SCREEN."
400 DATA P
410 DATA "  TO START A NEW P
ARAGRAPH ENTER THE LETTER @P
@ AS A SEPERATE DATA STATEME
NT, THEN INDENT YOUR TEXT ON
 YOUR NEXT NEXT DATA"
420 DATA "STATEMENT 2 OR 3 S
PACES (IF DESIRED).",P,"TO S
KIP LINES,",P,"JUST ENTER @P
@",P,"WHERE EVER YOU WANT TO
",P,"SKIP."
430 DATA P,"MAKE SURE THAT Y
OUR VERY LAST DATA STATEMENT
 IS @ZZZ@, AND JUST REPLACE
THESE DATA STATEMENTS WITH"
440 DATA "YOUR OWN.",P,"YOU'
LL ALSO FIND THIS ROUTINE IS
 MOST USEFUL WHEN CONCATENAT
ING STRINGS, E.G., @ELIZA@ T
YPE PROGRAMS-",P
450 DATA "AN EXAMPLE:",P,"A$
=@JACK AND JILL WENT UP@","B
$=@THE HILL TO FETCH A@","C$
=@PAIL OF WATER.@","D$=A$&B$
&C$&D$","PRINT D$",P
460 DATA "JACK AND JILL WENT
 UP THE HILL TO FETCH A PAIL
 OF WATER.",P,P,P,"HAPPY PRO
GRAMMING!"
470 DATA ZZZ
480 SUB WAIT
490 DISPLAY AT(24,8):"PRESS
ANY KEY"
500 CALL KEY(0,K,S):: IF S=0
 THEN 500 ELSE CALL CLEAR
510 SUBEND
```

Thank you, Julie and John. This is becoming one of the most useful routines on my utility disk. I was preparing a disk of PD programs for our UG library. Some of them needed extra instructions, so I typed them out on TI-Writer, so that people could run them off on their printer. Then I remembered that some folks don't have printers. So -

```
50 CALL CLEAR :: INPUT "FILE
NAME? DSK1."?:F$
60 DIM B$(150):: OPEN #1:"DS
K1."&F$,INPUT, DISPLAY ,VAR
IABLE 80
70 A=A+1 :: LINPUT #1:B$(A)
80 IF EOF(1)=1 THEN B$(A+1)=
"ZZZ" ELSE 70
```

and change line 170 to -

```
170 @=@+1 :: MESS$=B$(@)
```

And there you have a quickie program to check out those DIS/VAR 80 files that show up on your disks under filenames that you can't remember using.

MEMORY FULL IN LINE 32767

Jim S.

USING LOOPS AND SUBSCRIPTED VARIABLES

Richard J. Bailey
65A Church Street
Conic,N. H. 03867

Anytime you need to perform a similar operation several times you should consider loops as a way of making your program shorter. The use of subscripted variables and loops can go hand-in-hand as you will later see.

If you want to print "HI" on the screen 20 times you could do the following;

```
10 PRINT "HI"
20 PRINT "HI"
```

30 PRINT "HI" and so on for a total of 20 identical lines. To shorten it you could use;

```
10 I=I+1
20 PRINT "HI";I
30 IF I=20 THEN 50
40 GOTO 10
50 END
```

for a total of 5 lines. This version also prints the value I, indicating the number of times it has printed. Changing the value in line 30 from 20 to 80 allows you to print "HI" 80 times. A more elegant way is to use a loop as follows;

```
10 FOR I=1 TO 20
20 PRINT "HI";I
30 NEXT I
```

for a total of 3 lines. If this were done in extended basic it would reduce to 1 program line; 10 FOR I=1 TO 20 :: PRINT "HI";I :: NEXT I. Obviously this is the way to go. The loop starts at some specified value (in this case, 1) and executes every statement between the FOR and NEXT the specified number of times (in this case, 20). To create a table or one dimensional array of 12 values from DATA statements you could use;

```
110 FOR I=1 TO 12
120 READ A(I)
130 NEXT I
140 DATA 2,4,1,6,3,1,6,8,10,3,2,4
```

Running the program immediately gives "BAD SUBSCRIPT IN 120" because you haven't reserved enough "boxes" in memory to put your numbers in. Subscripted variables have to be dimensioned if their number exceed 10. Most basics automatically reserve 10 "boxes" whenever they encounter a subscripted variable while executing a program. For this routine to work you need to add; 100 DIM A(12) at the begining of the program before the line that uses the first occurance of the subscripted variable. The reason you don't want to dimension subscripted variables any larger than necessary is that these reserved "boxes" are memory you can't use for other purposes. In large programs you could get an "OUT OF MEMORY" error message (even though there's enough memory) if you've dimensioned your variables with much larger values than needed. In a large program with many subscripted variables you might be able to free up memory by dimensioning variables less than 10, wherever possible. To see the effects of dimensioning subscripted variables on memory try the following example. First type NEW [ENTER] then SIZE [ENTER] in the immediate mode using extended basic. Copy down the number or numbers that appear on the screen. These are the total memory available. Enter the following two lines;

```
100 DIM A(1000)
110 BREAK
```

Before running the program, type SIZE [ENTER] again and copy down the numbers. These numbers, subtracted from the total memory, indicate

how many "boxes" or bytes of memory the program lines use. Now type RUN [ENTER]. When the program breaks, type SIZE [ENTER] again and copy down the numbers. The difference between these figures and the total memory is how much memory the program requires after reserving memory space for 1000 subscripted variables plus the memory required for the program lines. You will find the program before it ran, used about 23 bytes but after running with the subscripted variable memory space allocated, over 8000 bytes were reserved even though the variable isn't used for anything in the program.

Back to loops. If there are no lines between the FOR and NEXT what you have is a time delay. You can also have nested loops or loops within loops.

```
10 FOR I=1 TO 10
 20 PRINT "ON"
 30 FOR J=1 TO 5
 40 PRINT "OFF"
 50 NEXT J
 60 FOR TD=1 TO 250
 70 NEXT TD
 80 NEXT I
```

Just make sure that the "FOR J" is followed by "NEXT J" and not "NEXT I". Improper nesting or crossed loops will cause your program to crash. The inner or "J" loop executes 5 times for every execution of the outer or "I" loop. "OFF" will be printed a total of 10x5 or 50 times with "ON" printed every fifth time after a time delay (TD) of about 1 second. Also note on debugging that if line 50 didn't exist you would get a "FOR-NEXT ERROR" or "FOR-NEXT NESTING IN 80" message with extended basic. These errors can be hard to find because if the computer doesn't find the expected "NEXT" statement, it lists the last line in the program as the problem in extended basic or doesn't give you a clue in regular basic.

In the beginning I said if things are repeated that you should consider loops. If you have several loops, there can be the possibility of combining them. For example;

```
10 FOR I=1 TO 12
20 READ D(I)
30 NEXT I
40 FOR J=1 TO 12
50 READ M(J)
60 NEXT J
70 FOR K=1 TO 12
80 PRINT D(K);M(K)
90 NEXT K
100 DATA 1,3.....etc
110 DATA 10,30...etc
```

is the same as:

```
10 FOR I=1 TO 12
20 READ D(I),M(I)
30 PRINT D(I);M(I)
40 NEXT I
50 DATA 1,10,3,30...etc
```

(note the data lines must be rearranged so items are read in the correct order)

Note that loops don't have to count up and they don't have to step by 1. 100 FOR I=0 TO 33 STEP 27 :: PRINT #1:TAB(I);"filename";::NEXT I executes twice, tabbing 6 spaces, printing, tabbing 33 spaces, and printing again on the same line.

```
100 FOR I=10 TO 0 STEP -1 :: PRINT I ::CALL SOUND(200,1400,4)
    :: FOR J=1 TO 250 :: NEXT J :: NEXT I :: CALL SOUND(1000
    ,110,0)
```

will count down from 10 to 0.

```
100 FOR I=2 TO 10 STEP 2 :: PRINT I :: NEXT I
```

prints 2,4,6,8,10

Also if the start and stop are identical the loop will execute once. If the stop value is less than the start and the step is positive, the loop will not execute. Using variable limits and steps that are improperly defined can cause problems. For I=A TO B STEP C with A=1,B=-2, and C=1, the computer will pretend the loop doesn't exist.

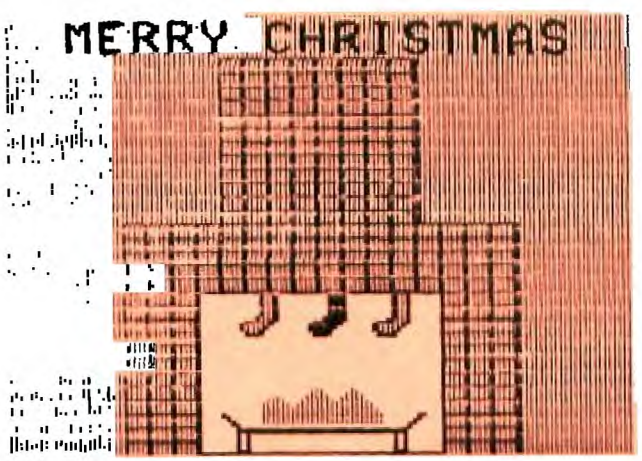
But now to put all these sometimes meaningless and boring pieces

together in a program. Please enter one of the following programs. One is for basic machines and the other for expanded machines with printers. These are practical (and timely) examples of how to use loops to greatly shorten a program. Note that only the string data with commas are enclosed in quotes. If no logical operations are being performed on the data, and no print seperators are included, quotes are not needed. If you use the printer version be sure to change the name of the print device to agree with your printer.

Also keep in mind if some of this looks like Greek that's exactly the way it looked to me at one time. The only way to learn is by doing, by trial and error, and a occasional curse.

```
100 REM ********************
110 REM *  BASIC 12 DAYS  *
120 REM *       BY       *
130 REM *RICHARD J. BAILEY*
140 REM *68A CHURCH STREET*
150 REM *GONIC, N.H. 03867*
160 REM ********************
170 DIM N$(12),S$(12)
180 FOR I=1 TO 12
190 READ N$(I),S$(I)
200 CALL CLEAR
210 PRINT "On the ";N$(I);"
day of Christmas"
220 PRINT "my true love gave
 to me-"
230 FOR J=I TO 1 STEP -1
240 PRINT S$(J)
250 NEXT J
260 FOR K=10-I/2 TO 1 STEP -
1
270 PRINT
280 NEXT K
290 FOR TD=1 TO 1800
300 NEXT TD
310 NEXT I
320 CALL CLEAR
330 DATA first,"a partridge
in a pear tree.",second,"two
 turtledoves, and",third,"th
ree french hens,",forth,"fou
r calling birds,"
340 DATA fifth,"five gold ri
ngs,",sixth,"six geese a-lay
ing,",seventh,"seven swans a
-swimming,",eighth,"eight ma
ids a-milking,"
350 DATA ninth,"nine drummer
s drumming,",tenth,"ten pipe
rs piping,",eleventh,"eleven
 ladies dancing,",twelth,"tw
elve lords a-leaping,"
```

```
100 !********************
110 !* X-BASIC 12 DAYS *
120 !*       BY         *
130 !*RICHARD J. BAILEY*
140 !*68A CHURCH STREET*
150 !*GONIC, N.H. 03867*
160 !********************
170 DIM N$(12),S$(12):: OPEN
 #1:"RS232.BA=2400.DA=8" ::
PRINT #1:CHR$(27);"E";CHR$(2
7);"G"
180 FOR I=1 TO 12 :: READ N$
(I),S$(I):: PRINT #1:TAB(27)
;"On the ";N$(I);" day of Ch
ristmas":TAB(27);"my true lo
ve gave to me-"
190 FOR J=I TO 1 STEP -1 ::
PRINT #1:TAB(27);S$(J):: NEX
T J :: PRINT #1:"" :: NEXT I
 :: PRINT #1:CHR$(27);"@" ::
 CLOSE #1
200 DATA first,"a partridge
in a pear tree.",second,"two
 turtledoves, and",third,"th
ree french hens,",forth,"fou
r calling birds,"
210 DATA fifth,"five gold ri
ngs,",sixth,"six geese a-lay
ing,",seventh,"seven swans a
-swimming,",eighth,"eight ma
ids a-milking,"
220 DATA ninth,"nine drummer
s drumming,",tenth,"ten pipe
rs piping,",eleventh,"eleven
 ladies dancing,",twelth,"tw
elve lords a-leaping,"
```



INSIDE: HOW TO FIX A DISK

NEW HAMPSHIRE 99'ERS USER GROUP, INC.
P.O. BOX 7199, HEIGHTS STATION
CONCORD, NEW HAMPSHIRE 03301

FIRST CLASS MAIL

[redacted]

EDMONTON USER'S GROUP
PO BOX 11983
EDMONTON, ALBERTA
CANADA T5J-3L1

[redacted]