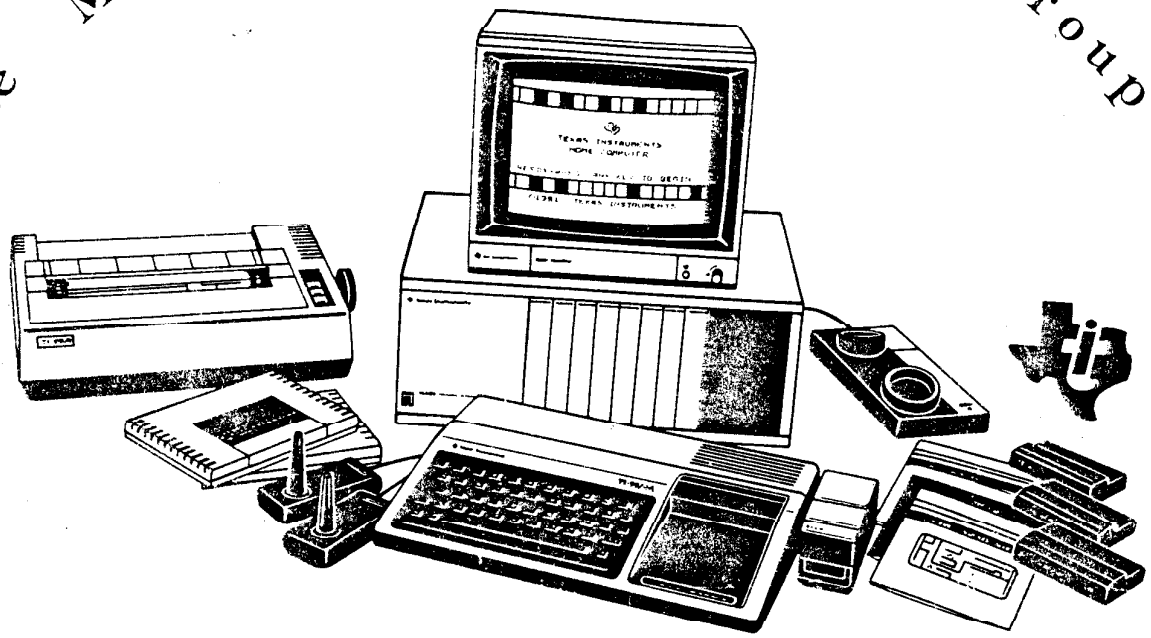
The Michiana 99/4A User's Group

911 Dover Drive
South Bend, IN 46614



TI-99/4A

The one to start with. The one to stay with.

MONTHLY NEWSLETTER

March 17, 1986

SOUTH BEND PUBLIC LIBRARY
ROGER B. FRANCIS BRANCH
52655 N. IRONWOOD RD.

for more information call:

DAVE FLOWERS
(219) 277-1990

THE MICHIANA 99/4A USER'S GROUP

MARCH 1986 NEWSLETTER VOLUME 4: NUMBER 3

CONTENTS:

This newsletter is free with membership in The Michiana 99/4A User's Group. We also make free exchanges of the newsletter with any other User's Group that wishes to reciprocate. Articles contained herein may be reprinted in another User's Group newsletter provided credit is given to the author and the source indicated herein.

Submissions of any kind - Questions and Answers, Programs, Program Reviews, Book Reviews, Advertisements - are invited and welcomed. Submission may be submitted at the monthly meetings or may be mailed to the User's Group at 52836 Searer Drive, South Bend, IN 46635. Submissions must be received by the second Monday of each month to be included in the current newsletter.

OFFICERS

President	Dave Flowers	277-1990
Vice-President	Roger Dooley	277-7306
Secretary	Larry Clough	272-9121
Treasurer	Ted Hatcher	471-3746 (Berrien Springs)

COMMITTEE MEMBERS

Equipment:	Dave Flowers	277-1990
Library:	Lois Wiley	293-8260 (Elkhart)
Meeting Notices:	Dennis Weigel	264-5615 (Elkhart)
Newsletter:	Mike Conway	291-4227
	Randy Devenport	291-2588
	Dave Flowers	277-1990
Program:	Mike Conway	291-4227
	Tom Kirk	293-0782 (Elkhart)
	Lois Wiley	293-8260 (Elkhart)

PRESIDENT'S COLUMN

Happy St. Patrick's Day! May the luck of the Irish be with us as we forge ahead looking for new ways to use our 99's.

Thanks to Ted Hatcher who gave us a demonstration of Millers Graphics **Advanced Diagnostics** at the February meeting. The utility allows you to read, print, or change individual sectors and look in between sectors. Ted even demonstrated how it can check the rotation speed of the disc drive. I must confess that it was a bit over my head but interesting nonetheless.

A plus from our newsletter exhange -- I've been trying to learn how to set printer commands to enlarge print, condense print, etc., but was not having much success in understanding my Gemini printer manual. So what should appear in the March 1986 newsletter of the Boise 99'ers Computer Club but a handy chart of printer commands which I am happy to reprint in this month's newsletter. Thank you, Jerry Hough, President of the Boise 99'ers.

No one from our Group, except unknowingly Gail Keb, offered any information for publication in this month's newsletter. Gail sent me a program for Jingle Bells with which she is having a problem. I'm not much into programming or debugging so I have decided to share her problem with the rest of you as an example of sharing problems as well as information. If you can solve Gail's problem, please give her a call. And, let me know also so that I can publish the solution in a future newsletter.

In this regard, I would like **YOU** to start thinking of this newsletter as a conduit -- something for **YOU** to use to share information and problems, ask questions, swap and sell, etc. Think about how **YOU** can start making our newsletter "user friendly."

A REALLY NIFTY TI-WRITER UPGRADE

Anyone who uses TI-Writer will want to consider a program by Paolo Bagnaresi. (His address is: Via J.F. Kennedy 17, 20097 San Donato Milanese, Italy. His phone number is 011-39-2-514.202 direct from the U.S.) He is asking $10 for it. Called BA-Writer, this program not only allows users to load TI-Writer using Extended BASIC, Editor/Assembler, Mini-Memory or TI-Writer, but provides an outstanding disk directory facility that is a marked improvement over the disk directory feature of TI-Writer itself. (TK-Writer and other loader programs allow users to load TI-Writer without the TI-Writer cartridge, but they do not support a disk-directory feature, which BA-Writer does.) The BA-Writer disk directory is super-fast and works out of the editor and the formatter. It will go through a double-sided, double-density diskette filled with 89 programs and files in less than 40 seconds. Of course, the TI-Writer files work flawlessly. Also, once you've loaded BA-Writer, you can pull the cartridge out of the console, and it will have no effect on the program. Even if you happen to exit the editor, for example, you may reload it instantaneously (literally), with the text file still intact. It works better than the Recover File feature of TI-Writer. With BA-Writer, you can throw away your TI-Writer cartridge. Now, if only we could do the same thing with Multiplan.

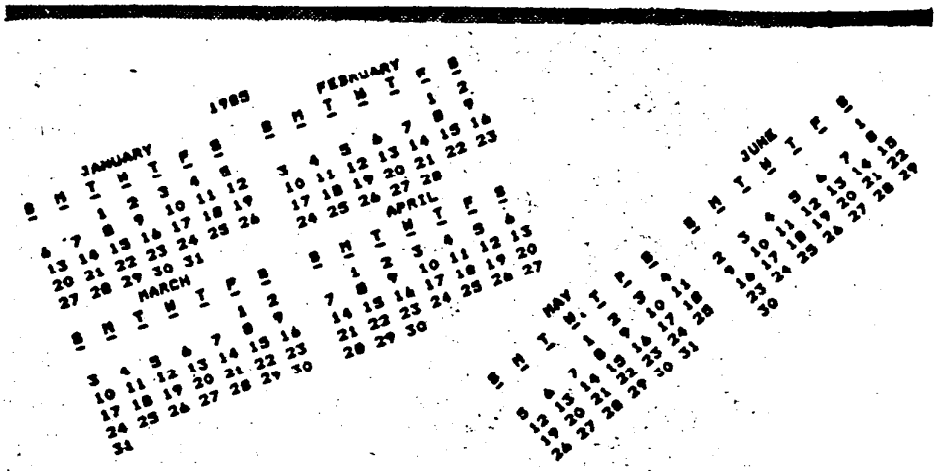
--FUTURE MEETINGS--
SCHEDULED THIRD MONDAY OF EACH MONTH AT
ROGER B. FRANCIS BRANCH
SOUTH BEND PUBLIC LIBRARY
52655 N. IRONWOOD ROAD
(JUST NORTH OF CLEVELAND ROAD)

APRIL 21
MAY 19
JUNE 16
JULY 21

Dates at a glance with TINY/CAL

Companies that produce calendars have one goal in mind: helping others keep track of days, weeks and months. It's a multi-million dollars business with a market that won't quit. Afterall, just about everyone has need for a calendar, every year.

Richard J. Bailey of Gonic, New Hampshire, is listing a number of programs as Freeware. Among them are several that produce and print calendars for any year from 1776 to 2099. These calendars are printed in a variety of sizes using Epson/TI type printers, including Gemini. The program listed here is called TINYCAL and produces, what else, a tiny calendar.



The program can probably be modified to operate using any dot-matrix printer that includes super/subscript characters. Although it is designed for RS232 operation, users may use parallel printers simply by changing the I/O characteristics in line 280. It is in line 280 that the super/subscript characters are accessed. This line may be used as the basis for "miniaturizing" printer output for many programs, including disk catalog programs.

The program requires Extended BASIC.

TINYCAL

```
100 !********************
110 !*      TINY        *
120 !*EPSON/TI CALENDER*
130 !*       BY         *
140 !*RICHARD J. BAILEY*
150 !*68A CHURCH STREET*
160 !*GONIC, N.H. 03867*
170 !********************
180 DIM T(12),D(12),MO$(12)
:: CALL CLEAR :: CALL SCREEN(
2):: FOR I=0 TO 14 :: CALL C
OLOR(I,16,2):: NEXT I
190 FOR I=1 TO 12 :: READ T(
I),D(I),MO$(I):: NEXT I
200 DATA 7,31,JANUARY,30,28,
FEBRUARY,8,31,MARCH,32,30,AP
RIL,9,31,MAY,32,30,JUNE
210 DATA 9,31,JULY,31,31,AUG
UST,6,30,SEPTEMBER,30,31,OCT
OBER,7,30,NOVEMBER,30,31,DEC
EMBER
220 DISPLAY AT(5,14):"TINY":
"      EPSON/T.I. CALENDAR":
"":"":"**THIS PROGRAM WILL P
RINT A":" CALENDAR FOR ANY
YEAR FROM":" 1776 TO 2099."
230 DISPLAY AT(13,1):"**SET
TOP OF FORM AND ENTER":" TH
E YEAR AS A FOUR DIGIT":" N
UMBER (ex. 1985) OR":"
JUST ENTER TO EXIT PROGRAM"
240 DISPLAY AT(19,1)BEEP:"**
ENTER CALENDAR YEAR" :: ACCE
PT AT(19,24)SIZE(4)VALIDATE(
DIGIT):Y$
250 IF Y$="" THEN CALL CLEAR
 :: END ELSE Y=VAL(Y$):: IF
Y<1776 OR Y>2099 THEN 240
260 IF INT(Y/4)*4=Y AND NOT(
INT(Y/100)*100=Y AND INT(Y/4
00)*400<>Y)THEN D(2)=29
270 DI=Y-1906+INT((Y-1901)/4
):: D(0)=DI+1-(INT(DI/7)*7)
280 M2=0 :: OPEN #1:"RS232.B
A=2400.DA=8" :: PRINT #1:CHR
$(27);"S";CHR$(1);CHR$(15);C
HR$(27);"3";CHR$(14);TAB(19)
;Y
290 FOR I=1 TO 12 STEP 2 ::
PRINT #1:TAB(T(I));MO$(I);TA
B(T(I+1));MO$(I+1)
300 J,K=1 :: A,M1=D(I-1)+M2
:: B,M2=M1+D(I)
310 PRINT #1:CHR$(27);"3";CH
R$(8);"S  M  T  W  T  F  S
  S  M  T  W  T  F  S":CHR$(
27);"3";CHR$(14);"-  -  -  -
  -  -  -  -  -  -  -  -
-  -"
320 IF J>D(I)THEN 330 :: IF
A>7 THEN A=A-7 :: GOTO 320 E
LSE PRINT #1:TAB(A*3-2);STR$
(J);:: IF A=7 THEN 330 ELSE
A=A+1 :: J=J+1 :: GOTO 320
330 IF K>D(I+1)THEN 340 :: I
F B>7 THEN B=B-7 :: GOTO 330
 ELSE PRINT #1:TAB(21+B*3);S
TR$(K);:: IF B=7 THEN 340 EL
SE B=B+1 :: K=K+1 :: GOTO 33
0
340 IF J>D(I)AND K>D(I+1)THE
N 350 ELSE PRINT #1:"" :: A=
A+1 :: B=B+1 :: J=J+1 :: K=K
+1 :: GOTO 320
350 PRINT #1:"" :: NEXT I ::
 PRINT #1:"":CHR$(27);"@" ::
 CLOSE #1 :: RESTORE :: GOTO
 190
```

TIPS FROM THE TIGERCUB

#27

TIGERCUB SOFTWARE
156 Collingwood Ave.
Columbus, OH 43213

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

>>>>>>NOW AVAILABLE<<<<<<<<<

Tips from the Tigercub VOLUME 2 The entire contents of Tips Nos. 15 through 24, with 64 routines and files, also $15.00 postpaid.

Or both for $27 postpaid.

>>>>ALSO NOW AVAILABLE<<<<<<

New Catalog #6, for $1 which is deductable from your first order. Describes 140 original programs for only $3 each (plus $1.50 per order for casette or disk, package and postage).

If you have my previous catalog, the following are now available in Extended Basic versions - Fast Addition Practice, Submarine Hunt, Rithmatik, Wawaland (also now available in Basic with Speech), Long Division Cryptograms, Miss Spell, Scrambulation, Bargraffer, Squinch, Dry Gulch, Name That Tune, Scrum, Midnight Trail, Nimbo, Kindertimes, Optical Illusion, Bazoo, Synonymy, Speeder Reader, Changeroo, Glunk, Fraction Math, Three Buckets Puzzle, Roman Numbers, Match A Patch, Kinderminus, I & E Spelling, Casting Out Nines, Haunted Graveyard, Spalling Teecher, Homonymy, Antonymy, Old -Timer Puzzle, Ten Thousand Sights, Mechanical Aptitude Test, Junior Speeder Reader, and Bars and Balls.

Due to reduced prices for disks and mailers, the PPM charge is now $1.50 for either disk or casette - BUT PLEASE BE SURE TO SPECIFY WHICH!

And my best seller - NUTS & BOLTS, a full disk of 100 (yes, I said 100) utility subprograms in MERGE format, ready for you to merge into your own programs. 13 type fonts, 14 text display routines, 9 wipes, 8 pauses, 3 programming aids, 9 data saving and reading routines, 5 graphics routines, 4 time and date, 6 music, 12 sorts and shuffles, 2 printer aids, 4 key and joystick, 4 math, 2 protection and 7 miscellaneous, plus a tutorial on subprograms. With documentation, example of using each subprogram. All for only $19.95 postpaid.

I have been receiving several requests to publicize freeware which is sometimes good but sometimes of doubtful usefulness, quality, originality or even legality! And "Freeware! Send $10 and initialized disk" is not freeware, it's somebody trying to get a free ad! So - no more freeware mentions! I am also not going to mention commercial products - after all, I'm publishing this at my own expense to promote my own software! However, I do owe a mention to Larry Hughes of Quality 99 Software, because in Tips #22 I recommended that disks with fractured files should not be copied with a quick copier. Larry informed that his trademarked QUICK-COPYer is the only program of its kind on the market that does un-fracture files. He sent along a copy to prove it, and it does just that. A very useful feature!

Now, here is the new, and final, version of the Tigercub Menu Loader.

```
100 !by A. Kludge/M. Gordon/
T. Boisseau/J. Peterson/etc.
Version #5, 9/85
110 CALL PEEK(8198,A):: IF A
<>170 THEN CALL INIT
120 OPTION BASE 1 :: DIM P$(
127),V(127,3):: CALL LOAD(-
31806,16):: ON ERROR 130 ::
GOTO 160
130 DISPLAY AT(12,9)ERASE AL
L:"I/O ERROR" :: RUN 100
140 @,@@,A,A$,B,C,D$,FLAG,I,
J,K,KD,KK,M,M$,N$,NN,P,P$,P6
$(),PP,PP$,Q$,S,ST,T$(),TT,V
T,V(,),W$,X,X$,Y,K2,S2
150 CALL LINK :: CALL PEEK :
: CALL KEY :: CALL SCREEN ::
 CALL COLOR :: CALL CLEAR ::
 CALL VCHAR :: CALL SOUND ::
 !@P-
160 CALL CLEAR :: CALL LOAD(
8196,63,248):: CALL LOAD(163
/6,6/,85,82,83,79,82,48,8)
170 CALL LOAD(12288,129,195,
126,165,129,153,102,60)
180 CALL LOAD(12296,2,0,3,24
0,2,1,48,0,2,2,0,8,4,32,32,3
6,4,91):: CALL LINK("CURSOR"
)
190 CALL CLEAR :: FOR S=1 TO
 14 :: CALL COLOR(S,7,16)::
NEXT S :: CALL COLOR(0,2,16)
200 T$(1)="d/f" :: T$(2)="d/
v" :: T$(3)="i/f" :: T$(4)="
i/v" :: T$(5)="pro" :: ON WA
RNING NEXT
210 IMAGE ###
220 IMAGE ###  Quit
230 IMAGE ###  Delete
240 IMAGE ###  Print
250 IMAGE ###  Rescan
260 CALL SCREEN(5):: CALL VC
HAR(1,31,1,96):: DISPLAY AT(
1,4):"TIGERCUB MENU LOADER"
270 ! IF YOU HAVE MORE THAN
ONE DISK DRIVE, DELETE THE !
 IN LINE 200 AND THE FIRST S
TATEMENT IN 210
280 ! DISPLAY AT(12,6):"DISK
? (1-3):" :: ACCEPT AT(12,19
)SIZE(-1)VALIDATE("123"):D$
:: D$="DSK"&D$&"."
290 D$="DSK1." :: OPEN #1:D$
,INPUT ,RELATIVE,INTERNAL ::
 INPUT #1:N$,A,J,K :: DISPLA
Y AT(1,2)SIZE(27):SEG$(D$,1,
4)&" - Diskname= "&N$;
300 DISPLAY AT(2,2):"Availab
le=";K;"Used=";J-K:" Prog Fi
lename Size  Type":"---- ---
------- ---- -------" :: I,V
T=0 :: TT=J-K
310 FOR X=1 TO 127 :: IF X/2
0<>INT(X/20)THEN 340
320 DISPLAY AT(24,1):"Choice
? Enter for more 0" :: ACCEP
T AT(24,24)VALIDATE(DIGIT)SI
ZE(-3):K :: IF K=0 THEN 330
:: IF K>0 AND K<NN+1 THEN 60
0 ELSE 320
330 X=1
340 I=I+1 :: IF I>127 THEN K
=X :: GOTO 510
350 INPUT #1:P$,A,J,B :: NN=
NN+1
360 IF LEN(P$)=0 THEN 430
370 DISPLAY AT(X+4,1):USING
210:NN :: DISPLAY AT(X+4,5):
P$ :: P6$(NN)=P$ :: DISPLAY
AT(X+4,16):USING 210:J :: DI
SPLAY AT(X+4,20):T$(ABS(A))
380 V(NN,1)=A :: V(NN,2)=ABS
(B):: V(NN,3)=J
390 X$="  "&STR$(B):: DISPLA
Y AT(X+4,24):SEG$(X$,LEN(X$)
-2,3):: VT=VT+J
400 IF A>0 THEN 410 :: DISPL
AY AT(X+4,28):"Y"
410 CALL KEY(0,KK,ST):: IF S
T=0 THEN 420 :: FLAG=1 :: GO
TO 430
420 NEXT X
430 DISPLAY AT(X+4,1):USING
220:NN :: DISPLAY AT(X+5,1):
USING 230:NN+1
440 IF VT=TT OR FLAG=1 THEN
460 :: DISPLAY AT(2,25)SIZE(
4):VT
450 FOR @=1 TO 10 :: DISPLAY
 AT(2,25)SIZE(1):CHR$(30)::
DISPLAY AT(2,25)SIZE(1):" "
:: CALL SOUND(-99,110,0,-4,0
):: NEXT @
460 IF FLAG=1 THEN 470 :: DI
SPLAY AT(X+4,13):USING 240:N
N+2 :: DISPLAY AT(X+5,13):US
```

```
IN6 250:NN+3
470 DISPLAY AT(X+6,1):"    C
hoice?" :: ACCEPT AT(X+6,16)
SIZE(-3)VALIDATE(DIGIT):K
480 IF FLAG=1 THEN 500
490 IF K=NN+2 THEN 840 ELSE
IF K=NN+3 THEN CLOSE #1 :: N
N=0 :: GOTO 190
500 IF K<>NN AND K<>NN+1 THE
N 590
510 IF K=NN THEN CALL CLEAR
:: CLOSE #1 :: END
520 DISPLAY AT(X+5,12)SIZE(1
2):" #?" :: ACCEPT AT(X+5,15
)SIZE(2)VALIDATE(DIGIT):KD :
: IF KD<1 OR KD>NN THEN 520
530 IF V(KD,1)>0 THEN 550
540 FOR J=1 TO 10 :: DISPLAY
 AT(11,1):" ":"  PROTECTED -
 CANNOT DELETE":" " :: DISPL
AY AT(12,1):" " :: NEXT J ::
 GOTO 570
550 DISPLAY AT(X+6,1)SIZE(27
)BEEP:" Verify - Delete ";P6
$(KD);"?" :: DISPLAY AT(X+6,
28)SIZE(1):"Y" :: ACCEPT AT(
X+6,28)SIZE(-1)VALIDATE("YN"
):Q$ :: IF Q$<>"Y" THEN 570
560 DELETE D$&P6$(KD)
570 CLOSE #1
580 CALL VCHAR(1,3,32,672)::
 NN=0 :: X=0 :: FLAG=0 :: GO
TO 260
590 IF K<1 OR K>127 OR LEN(P
6$(K))=0 THEN 430
600 IF ABS(V(K,1))=5 OR ABS(
V(K,1))=4 AND V(K,2)=254 THE
N 640
610 DISPLAY AT(12,1)ERASE AL
L:"Print to ? S": :"(P)rinte
r?":"(S)creen?" :: ACCEPT AT
(12,12)SIZE(-1)VALIDATE("PS"
):Q$ :: IF Q$="S" THEN PP=0
:: GOTO 630
620 DISPLAY AT(12,1)ERASE AL
L:"PRINTER? PIO" :: ACCEPT A
T(12,10)SIZE(-18):P$ :: OPEN
 #3:P$ :: PP=3
630 CALL CLEAR :: CALL SCREE
N(16):: ON ABS(V(K,1))GOTO 6
80,690,750,760
640 CLOSE #1 :: IF SEG$(P6$(
K),LEN(P6$(K)),1)="*" THEN D
ISPLAY AT(12,1)ERASE ALL:"RE
TURN TO BASIC AND LOAD BY":"
TYPING OLD ";D$&P6$(K):: STO
P
650 CALL PEEK(-31952,A,B)::
CALL PEEK(A*256+B-65534,A,B)
:: C=A*256+B-65534 :: A$=D$&
P6$(K):: CALL LOAD(C,LEN(A$)
)
660 FOR I=1 TO LEN(A$):: CAL
L LOAD(C+I,ASC(SEG$(A$,I,1))
):: NEXT I :: CALL LOAD(C+I,
0)
670 CALL VCHAR(1,3,32,672)::
 CALL SCREEN(8):: FOR S=0 TO
 14 :: CALL COLOR(S,2,1):: N
EXT S :: DISPLAY AT(12,2):"L
OADING ";A$ :: GOTO 900
680 OPEN #2:D$&P6$(K),INPUT
,FIXED :: GOTO 700
690 OPEN #2:D$&P6$(K),INPUT
700 LINPUT #2:W$ :: PRINT #P
P:W$ :: IF EOF(2)THEN 730
710 CALL KEY(0,K,S):: IF S=0
 THEN 700
720 CALL KEY(0,K2,S2):: IF S
2<1 THEN 720 ELSE 700
730 CLOSE #1 :: CLOSE #2 ::
PRINT "   >>>press any key<<
<" :: IF Q$="P" THEN CLOSE #
3
740 CALL KEY(0,K,ST):: IF ST
<1 THEN 740 ELSE 580
750 OPEN #2:D$&P6$(K),INPUT
,INTERNAL,FIXED :: J=0 :: GO
TO 770
760 OPEN #2:D$&P6$(K),INPUT
,INTERNAL :: J=0
770 IF EOF(2)=1 THEN 730 ::
J=J+1 :: INPUT #2:M$ :: IF L
EN(M$)=8 THEN 790
780 PRINT #PP:M$ :: GOTO 820
790 FOR Y=1 TO 8 :: @@=ASC(S
EG$(M$,Y,1)):: IF @@<32 OR @
@>127 THEN 810
800 NEXT Y :: GOTO 780
810 RESTORE #2 :: FOR X=1 TO
 J-1 :: INPUT #2:M$ :: NEXT
X :: INPUT #2:M :: PRINT #PP
:M
820 CALL KEY(0,K,S):: IF S=0
 THEN 770
830 CALL KEY(0,K2,S2):: IF S
2<1 THEN 830 ELSE 770
840 DISPLAY AT(24,1):"PRINTE
R NAME? PIO" :: ACCEPT AT(24
,15)SIZE(-14):PP$ :: OPEN #2
:PP$ :: PRINT #2:SEG$(D$,1,4
)&" - Diskname= "&N$
850 PRINT #2:RPT$("*",28):"A
vailable=";358-VT;"Used=";VT
:RPT$("~",28)
860 PRINT #2:"FILENAME  SIZE
   TYPE":RPT$("_",28)
870 FOR P=1 TO NN-1 :: PRINT
 #2:P6$(P);TAB(15);V(P,3);TA
B(20);T$(ABS(V(P,1)));TAB(25
);V(P,2):: NEXT P :: CLOSE #
2
880 DISPLAY AT(12,3)ERASE AL
L:"(P) to print again":" (R
) to rescan":" (Q) to quit"
890 ACCEPT AT(15,4)VALIDATE(
"PQR")SIZE(-1)BEEP:Q$ :: IF
Q$="P" THEN 840 :: CLOSE #1
:: NN=0 :: IF Q$="R" THEN 19
0 ELSE END
900 RUN "DSKX.1234567890"
```

This version turns off the Quit key, restarts itself rather than crashing on an I/O error, and has pre-scan for faster start-up. It displays disk name, sectors available and sectors presumably used - it also totals up actual sectors used and sounds a warning if any sectors are not accounted for.

It lists up to 127 programs and files by number, filename, number of sectors, program or file type, file record length, and write-protection. It will stop for menu selection on any keypress or at the end of each screen, continuing on Enter. It will load and run any program that can run from Extended Basic, displaying its filename while loading. If the filename ends in an asterisk, it will warn you to return to Basic. It will delete any unprotected program or file, after first requiring verification by filename, or will inform you if the file is protected. It will read any readable file, including internal numeric, and list it to screen or printer. It will dump a catalog of the disk to your printer, and it will offer the option of quitting or rescanning the disk or another disk. And it's free, I don't even want a freeware donation - but I would appreciate if you would take a look at my catalog and see if, somewhere among those 140 programs, there might be something you would be willing to pay $3 for? The Menu Loader is included as a bonus on every disk I sell!

```
100 CALL CLEAR :: RANDOMIZE
:: DISPLAY AT(3,4):"TIGERCUB
 MATH PUZZLE"
110 DISPLAY AT(6,1):"Insert
+, -, * (multiply) OR / (div
ide) between the  digits
to equal the total": :"Type
Q to give up"
120 DISPLAY AT(12,1):"Level
1 or 2?" :: ACCEPT AT(12,15)
VALIDATE("12"):L$
130 T,X=INT(9*RND+1):: M$=ST
R$(X):: Z$=M$&" "
140 FOR J=1 TO 4 :: Y(J)=INT
(9*RND+1):: Z=INT(4*RND+1)::
 ON Z GOSUB 240,250,260,270
:: Z$=Z$&STR$(Y(J))&" " :: N
EXT J
150 IF L$="1" AND T<>INT(T)T
HEN 130 :: Z$=Z$&"="&STR$(T)
160 DISPLAY AT(12,1):Z$ :: D
ISPLAY AT(18,1):" " :: DISPL
AY AT(20,1):" " :: DISPLAY A
T(22,1):" "
170 P=2 :: FOR J=1 TO 4 :: A
CCEPT AT(12,P)VALIDATE("Q+-*
/")SIZE(1):S$
180 IF S$="Q" THEN 200 ELSE
IF S$="+" THEN X=X+Y(J)ELSE
IF S$="-" THEN X=X-Y(J)ELSE
IF S$="*" THEN X=X*Y(J)ELSE
X=X/Y(J)
190 P=P+2 :: NEXT J :: IF X=
T THEN 230 :: DISPLAY AT(18,
1):"WRONG!"
200 DISPLAY AT(20,1):"ANSWER
 IS ";M$
210 DISPLAY AT(22,1):"PRESS
ANY KEY"
220 CALL KEY(0,K,ST):: IF ST
<1 THEN 220 :: GOTO 130
230 DISPLAY AT(18,1):"RIGHT!
" :: GOTO 210
240 M$=M$&"+"&STR$(Y(J)):: T
=T+Y(J):: RETURN
250 M$=M$&"-"&STR$(Y(J)):: T
=T-Y(J):: RETURN
260 M$=M$&"*"&STR$(Y(J)):: T
=T*Y(J):: RETURN
270 M$=M$&"/"&STR$(Y(J)):: T
=T/Y(J):: RETURN
```

Enjoy!

Jim Peterson

TIPS FROM THE TIGERCUB

#28

TIGERCUB SOFTWARE
156 Collingwood Ave.
Columbus, OH 43213

NUTS & BOLTS DISK No. 2 is now ready, and I think it's better than the first one. It contains 188 utility subprograms in merge format, including many new character fonts and screen display routines as well as 2-dimensional array sorts, variable line numbers in GOSUB, GOTO and RESTORE, on-screen editing and much, much more. The price is $19.95 postpaid, or you can order both Nuts & Bolts disks for $37 ppd.

And I have put together 18 different collection disks each containing 5 or 6 of my catalog programs for just $12 postpaid. The programs on each disk are all of the same category, and I have filled up the rest of the disk with public domain programs of the same category, as a bonus.
I want to make it very plain that I am NOT - repeat, NOT - selling public domain programs! My own programs on these disks are offered at a great discount and the public domain programs are just thrown in for free! Together with this issue of the Tips I am mailing to each user's group a copy of my catalog #6 with an added page describing these new offerings, and a rebate offer to user's groups.

My catalog will be sent to individuals for $1, which is deductable from your first order. If you already have my catalog #6, the added page will be sent to you free on request.

My full disk collections will now be available to bona-fide retailers at standard wholesale prices. Inquiries on your letterhead are invited.

And so, on to old business. Yes, I know that RESequencing a program does not resequence references to line numbers in REMs. I just forgot! In line 278 of the Menu Loader in Tips #27, the reference should be to lines 288 and 298, of course.
While programming the file reader in that menu loader, I ran into a peculiarity of the TI-99/4A that surprised most of the expert programmers whom I called for help. When you "read blind" you must read everything as a string, because attempting to read a string as numeric will crash the program. This is no problem with DISPLAY files - but when I tried it with INTERNAL files, I got the strangest garbage! My solution (not quite fool-proof) was to identify a record as numeric if it was 8 bytes long and contained an ASCII out of printable range, and then RESTORE the file, read back to that point and re-read it as numeric. Not very efficient!
The following routine will save a numeric input in an internal file, read it back out as a string, show you the way it was saved, and then attempt to translate it back to numeric. It works for positive and negative integers or non-integers of not less than -99, but not for less than that.

```
100 INPUT X :: OPEN #1:"DSK1
.TEST",INTERNAL,OUTPUT :: PR
INT #1:X :: CLOSE #1
110 OPEN #1:"DSK1.TEST",INTE
RNAL,INPUT :: INPUT #1:A$ ::
 PRINT A$ :: CLOSE #1
120 FOR J=1 TO 8 :: PRINT AS
C(SEG$(A$,J,1));:: NEXT J
130 FOR J=1 TO 8 :: A(J)=ASC
(SEG$(A$,J,1)):: NEXT J
140 X=A(1)-63 :: IF X<73 THE
N 150
142 X=192-A(1):: N$="-" :: F
OR J=2 TO X+1 :: N$=N$&STR$(
256-A(J)):: NEXT J :: GOTO 1
60
150 FOR J=2 TO X+1 :: N$=N$&
STR$(A(J)):: NEXT J
160 IF A(J)<>0 THEN N$=N$&".
"&STR$(A(J))
170 J=J+1 :: IF A(J)<>0 THEN
 N$=N$&STR$(A(J)):: GOTO 170
180 N=VAL(N$):: N$="" :: PRI
NT N :: GOTO 100
```

So, here is another Tigercub Challenge! Can you fix it? Let's HEAR from you this time!

Another problem that I ran into was in recovering from an I/O error. When ON ERROR is used to prevent crashing on such an error, the file is "ajar" - you can't close it and you can't open it. My solution was to simply RUN the program again - and this will show you how the pre-scan speeds that up. Since then, I have learned of three other ways. The method described in the Sydney (Australia) news-letter is a bit complicated, but Irwin Hott gave me a simple solution - just increment the file number! Works fine if you don't increment it into the number of another open file on the disk. Chuck Grimes gave me an even better way - open and close anything else, even "PIO"! Example -

```
100 ON ERROR 110 :: OPEN #1:
"DSK1.TEST",OUTPUT :: PRINT
"CONTINUE PROGRAM" :: END
110 OPEN #1:"PIO" :: CLOSE #
1 :: PRINT "I/O ERROR":"CHEC
K DISK AND DRIVE":"THEN PRES
S ANY KEY" :: ON ERROR STOP
120 CALL KEY(0,K,S):: IF S=0
 THEN 120 ELSE 100
```

There is a reason for that ON ERROR STOP, and it's why I don't use ON ERROR if I can avoid it. When an error occurs, the program goes to the line number specified by the last open ON ERROR statement, takes whatever action is directed by that line, and RETURNs as directed. If the error was not one that you expected to happpen, the results can be very confusing!
For that reason, when you set out to modify a program, the first thing you should do is delete, temporarily, all the ON ERROR statements. The next thing you should do, if the program has a routine to turn off the pre-scan, is to disable that. Otherwise, you will be driven crazy by invalid SYNTAX ERROR messages and other strange happenings.
The third thing you should do is to make a list of all the lines that a GOTO or GOSUB goes to, so you don't delete or change them. And here is a program to do just that for you -

```
100 !GO-SEARCH by Jim Peters
on searches a MERGE format f
ile, finds all line numbers
containing a jump, sorts int
o "to" line number sequence,
110 !prints "to" line number
, statement (GO, GOTO or GOS
UB) and "from" line number
120 DIM C(200):: A=1 :: GO$(
1)="GO" :: GO$(2)="GOTO" ::
GO$(3)="GOSUB"
130 INPUT "FILENAME? DSK1.":
F$
140 OPEN #1:"DSK1."&F$,INPUT
 ,VARIABLE 163 :: OPEN #2:"P
IO"
150 LINPUT #1:A$
160 IF POS(A$,CHR$(133),1)=0
 AND POS(A$,CHR$(134),1)=0 A
ND POS(A$,CHR$(135),1)=0 THE
N 210
170 LN=ASC(SEG$(A$,1,1))*256
+ASC(SEG$(A$,2,1)):: T=133 :
: P=1
180 G$=CHR$(T):: X=POS(A$,G$
```

```
,P):: IF X=0 THEN 200 :: LRE
F=ASC(SEG$(A$,X+2,1))*256+AS
C(SEG$(A$,X+3,1))!:: PRINT #
2:LN;GO$(T-132);LREF :: P=X+
1 :: GOTO 180
190 C$=STR$(LREF)&","&STR$(L
N)&STR$(T-132):: C(A)=VAL(C$
):: A=A+1 :: P=X+1 :: GOTO 1
80
200 IF G$=CHR$(135)THEN 210
:: T=T+1 :: P=1 :: GOTO 180
210 IF EOF(1)THEN CLOSE #1 :
: GOTO 220 :: ELSE 150
220 A=A-1 :: CALL LONGSHELLN
(A,C())
230 FOR J=1 TO A :: A$=STR$(
C(J)):: X=POS(A$,".",1):: Y=
VAL(SEG$(A$,LEN(A$),1)):: A$
=SEG$(A$,1,LEN(A$)-1)
240 PRINT #2:SEG$(A$,1,X-1);
TAB(7);GO$(Y);" FROM ";TAB(2
1);SEG$(A$,X+1,LEN(A$)):: NE
XT J
250 SUB LONGSHELLN(N,NN())
260 D=N
270 D=INT(D/3)+1 :: FOR I=1
TO N-D :: IF NN(I)<=NN(I+D)T
HEN 300 :: T=NN(I+D):: J=I
280 NN(J+D)=NN(J):: J=J-D ::
 IF J<1 THEN 290 :: IF T<NN(
J)THEN 280
290 NN(J+D)=T
300 NEXT I
310 IF D>1 THEN 270
320 SUBEND
```

According to the User's Reference Guide that came with your computer, if you open a file without specifying INPUT, OUTPUT, UPDATE or APPEND, the computer will assume the UPDATE mode as the default and "UPDATE files may be both read and written. The usual processing is to read a record, change it in some way, and then write the altered record back out on the file." This is a very dangerous bit of misinformation! It is true only if you are using RELATIVE files with the REC clause. In any other case, the first record you write to the file will become the record FOLLOWING the last record you read, and it will also become the LAST record in the file - any records beyond that point will be lost!

The moral of the story - get in the habit of NEVER opening a file without specifying the mode. The only way to update a sequential file is to read it ALL into an array, update it, and then write it back to the file.

I reviewed hundreds of programs, in my PD library of about 2600, in order to select some of the best to fill up the collection disks. Often they needed only a few minor changes to greatly improve them.

One frequent flaw was in interpreting the status of CALL KEY. The User's Reference Guide says that a status variable of -1 means that "the same key was pressed during the performance of CALL KEY as was pressed during the previous performance." This is misleading. It actually means that the same key is STILL BEING pressed. Try this -

```
100 DISPLAY AT(12,1)ERASE AL
L:"TYPE YOUR NAME" :: R=12 :
: C=3
110 CALL KEY(0,K,S):: IF S=0
 THEN 110 :: DISPLAY AT(R,C)
:CHR$(K):: C=C+1 :: GOTO 110
```

Difficult to type without unwanted repetition of letters? Now try changing the S=0 to S<1 !

IF S<1 (if S is less than 1) means that if no key is pressed (S=0) or if the same key is still being held down (S=-1) then CALL KEY again.

Another frequent flaw is

```
INPUT "WANT TO PLAY AGAIN? "
:Q$ :: IF Q$<>"Y" THEN END
```

- or, more professionally programmed, IF SEG$(Q$,1,1)<>"Y" THEN....,

which will accept either "Y" or "YES" as a reply. The problem is still that this question is often asked at the end of a joystick game, for which the Alpha Lock will be unlocked - and a response of a lower case "y" then terminates the program! One solution is to precede the INPUT with a dummy CALL KEY(3,K,S), which will cause any subsequent upper case CALL KEY, INPUT, LINPUT or ACCEPT AT response to be read as lower case until you turn it off with CALL KEY(5,K,S).

Here's one that does nothing except look pretty.

```
100 DISPLAY AT(3,8)ERASE ALL
:"COLORSQUARES" :: DISPLAY A
T(8,1):"Select option 1, 2 o
r 3" ! by Jim Peterson, Tige
rcub Software
110 CALL KEY(0,K,ST):: IF ST
=0 OR K<49 OR K>51 THEN 110
:: ON K-48 GOTO 150,120,130
120 FOR CH=38 TO 142 STEP 8
:: CALL CHAR(CH,RPT$("A55A",
4)):: NEXT CH :: GOTO 150
130 FOR CH=38 TO 142 STEP 8
:: FOR L=1 TO 4 :: RANDOMIZE
 :: X$=SEG$("0018243C425A667
E8199A5BDC3DBE7FF",INT(16*RN
D+1)*2-1,2)
140 B$=B$&X$ :: C$=X$&C$ ::
NEXT L :: CALL CHAR(CH,B$&C$
):: B$,C$=NUL$ :: NEXT CH
150 CALL CLEAR :: RANDOMIZE
:: FOR SET=0-(K>49)TO 14 ::
CALL COLOR(SET,SET+2+(K>49),
SET+2):: NEXT SET
160 Y=INT(4*RND+3):: R=INT(1
2*RND+1):: R2=25-R-Y :: C=IN
T(7*RND+7):: C2=32-C-Y :: IF
 K=49 THEN X=INT(14*RND+1)*8
+22 ELSE X=INT(13*RND+1)*8+3
0
170 FOR T=R TO R+Y :: CALL H
CHAR(T,C,X,Y):: CALL HCHAR(T
,C2,X,Y):: NEXT T
180 FOR T=R2 TO R2+Y :: CALL
 HCHAR(T,C,X,Y):: CALL HCHAR
(T,C2,X,Y):: NEXT T :: GOTO
160
```

The asterisk on the Gemini printer looks rather like a bug squashed side- ways, and it was confusing some folks in the condensed print of my newsletter, so I improved it with this -

```
150 PRINT #2:CHR$(27);CHR$(4
2);CHR$(1);CHR$(42);CHR$(0);
CHR$(8);CHR$(34);CHR$(8);CHR
$(0);CHR$(62);CHR$(0);CHR$(8
);CHR$(34);CHR$(8);
```

And at the same time I improved the slashed zero -

```
140 PRINT #2:CHR$(27);CHR$(4
2);CHR$(1);CHR$(48);CHR$(0);
CHR$(64);CHR$(30);CHR$(96);C
HR$(17);CHR$(72);CHR$(5);CHR
$(66);CHR$(61);CHR$(0);
```

```
90 !THIS WON'T WORK, WILL IT
?
100 DISPLAY AT(9999,9999)ERA
SE ALL:SEG$("CAN'T DO THAT!"
,1,3)&SEG$("CAN'T DO THAT!",
6,8)
```

If the Tigercub Math Puzzle in Tips #27 was a bit too tough, these changes will add a couple of easier levels.

```
105 DISPLAY AT(6,1):"Level 1
, 2, 3 or 4?" :: ACCEPT AT(6
,21)VALIDATE("1234"):L$ :: L
=VAL(L$)
106 IF L<3 THEN M$="Insert +
, -, or * (multiply)" ELSE M
$="Insert +, -, * (multiply)
 or / (divide)"
110 DISPLAY AT(5,1):M$;" bet
ween the digits":" to equal
the total": :"Type 0 to give
 up"
120 ! **DELETED LINE **
130 DISPLAY AT(12,1):" " ::
T,X=INT(9*RND+1):: M$=STR$(X
):: Z$=M$&" "
140 FOR J=1 TO 4 :: Y(J)=INT
(9*RND+1):: @=3+ABS(L>2):: Z
=INT(@*RND+1):: ON Z GOSUB 2
40,250,260,270 :: Z$=Z$&STR$
(Y(J))&" " :: NEXT J
150 IF L/2<>INT(L/2)AND T<>I
NT(T)THEN 130 :: Z$=Z$&"="&S
TR$(T)
```

MEMORY FULL

Jim Peterson

User Notes

Plotting circles

The question was simple: How do you draw a circle in BASIC? According to the Cin-Day (Ohio) Users Group, you can do it on paper or use the following program to plot it using CALL HCHARs. While the program doesn't actually draw a "circle," it does calculate and draw the closest approximation based on X,Y and radius coordinates. And it runs in BASIC with nothing added.

RC equals the center row of the circle. CC equals the center column of the circle and "radius" equals the distance in blocks or tiles on the screen of the circumference from the row and column center of the circle. Thus, you would enter the following numbers when prompted: 10, 12, 16 for radius, RC and CC, respectively. Row 12 and column 16 are the approximate center of the screen.

```
100 CALL CLEAR
110 INPUT "RADIUS,RC,CC? ";R
ADIUS,RC,CC
120 CALL HCHAR(1,1,32,704)
130 FOR X=-RADIUS TO RADIUS
STEP 1/RADIUS
140 R=X+RC
150 C=SQR(RADIUS^2-X^2)+CC
160 IF (R<1)+(R>24)THEN 220
170 IF (C<1)+(C>32)THEN 190
180 CALL HCHAR(R,C,42)
190 C=2*CC-C
200 IF (C<1)+(C>32)THEN 220
210 CALL HCHAR(R,C,42)
220 NEXT X
230 GOTO 110
```

Retrieving, reusing subroutines

You know how it is: you've got a 200 line program and you'd really like to save 15 lines of it as a subroutine but you wish there was an easier way than retyping the 15 lines or deleting the other 185 lines one by one.

Fortunately there is, according to George Steffen of the Los Angeles 99ers users group. Writing in the group's newsletter, Steffen provides a six-line program that does the job so well that you may decide to go back and extract subroutines from an entire library of programs just to make up for all the tedium you've had to put up with in the past.

The program is meant to be MERGEd into the program—which is the reason the program uses such low line numbers—you wish to extract the subroutine from, so, after saving it as a program, save it again in the MERGE format: SAVE DSKX.FILENAME, MERGE. Now, load the program from which you want to extract a subroutine—any group of consecutive program lines will do—and and then MERGE the subroutine extractor into it: MERGE DSKX.FILENAME. Enter RUN. You will be prompted for the starting and ending lines you wish to extract. Having done so, the program will do its job.

The proof of the job comes after the "READY" sign appears. List the program. You should see only the lines that you want to preserve. Now, save these to disk.

One caveat: it is suggested that you save the preserved lines in a MERGE format to start, because the lines that you sought to delete actually are still there. However, we found that the extracted lines can also be saved as a program. The deleted lines did not reappear in either case.

This program requires Extended BASIC, a disk system and memory expansion.

```
1 CALL CLEAR :: CALL INIT ::
 INPUT "LINE NUMBERS OF ROUT
INE TO BE SAVED: FIRST,LAST?
 ";L,M :: G=256 :: CALL PEEK
(-31952,H,I,J,K)
2 C=INT(M/G):: D=M-C*G :: F=
(J-G)*G+K :: FOR E=(H-G)*G+I
 TO F STEP 4 :: CALL PEEK(E,
A,B):: IF A=C AND B=D THEN 4
3 NEXT E :: PRINT :"LINE";0;
"NOT FOUND!" :: STOP !@P-
4 H=INT(E/G):: I=E-(G*H):: H
=H+G :: C=INT(L/G):: D=L-C*G
 :: FOR E=E+4 TO F STEP 4 ::
 CALL PEEK(E,A,B):: IF A=C A
ND B=D THEN 6 !@P-
5 NEXT E :: PRINT :"LINE";N;
"NOT FOUND!" :: STOP !@P-
6 E=E+3 :: J=INT(E/G):: K=E-
(G*J):: J=J+G :: CALL LOAD(-
31952,H,I,J,K):: STOP !@P-
```

Atari into TI

Who'd want to make that change, right? Well, we're not suggesting that we'd rather have an Atari than a TI (we wouldn't) but there may be some BASIC programs written for the Atari—or Apple, Commodore, TRS80, etc.—that would be worth having on the TI. Unfortunately, while BASIC for each of these machines has a lot in common, none is what you could call "transportable" without a little (sometimes a lot) of fiddling.

Gene Thomas, of the Jacksonville Users Group of Jacksonville, Arkansas, published a list of some the characteristics of various dialects of BASIC. He notes, for example, that the following items are virtually identical in all brands of BASIC: ABS, ASC, CHR$, DATA, DIM, END, GOTO, GOSUB, INT, LET, PRINT, READ, REM, RESTORE, RETURN, SQR, STR$, DEF, IF-THEN AND VAL.

But he didn't stop there. He also offers a list of some of the statements that you can expect to be different and their TI equivalents. Here they are:

CLS—CALL CLEAR
CLEAR—NEW(Not used within TI programs.)
CHANGE—ASC & CHR$
CINT, FIX—INT
CLG—LOG (Base 10)
CLOAD—Open cassette file and load
COLOR—CALL COLOR
CSAVE—Open cassette file and save
DEFINT—DEF"INT"(Declare DEF statement numerals to be integers.)
DEFSNG—May be ignored
DEFSTR—May be ignored
DLOAD—Open disk file and load
DSAVE—Open disk file and save
SET, DOT—CALL HCHAR, VCHAR
EQ—Equal sign

(Please turn to Page 50)

User Notes

(Continued from Page 49)

FRE—May be ignored
GET, INKEY$—INPUT, CALL KEY
GOSUB-OF—ON GOSUB
GOTO-OF—ON GOTO
INSTR—SEG$
LLIST, LPRINT—LIST, output to printer
**MID$**—SEG$
LEFT$(S$,N)—SEG$(S$,1,N)
RIGHT$(S$,N)—S = LEN(S$/(N + 1)) :: SEG$(S$,S,N)
RND(N)—INT(RND*N) + 1
PRINT USING—PRINT, DISPLAY USING
**STRING$(10,65)**—S$ = RPT$(65,10) :: PRINT S$
WAIT—FOR-NEXT delay loop
?—PRINT

Freeware update

SUPER-MAIL

Super-Mail consists of three programs. A LOAD program serves as the main menu for selecting either telephone directory or mailing label functions. The Telephone/address program allows for adding to and outputting a telephone-type directory. Included is a sort routine as well as a recall function. The mailing label program prints single-column labels using 3 1/2-inch by 15/16-inch labels. It prints single labels or the entire contents of the telephone/address directory. Send $5 to cover cost of postage, packaging and diskette to Henry J. Beck, 609 County Home Rd., Lexington, NC 27292.

SIDE*PRINT FOR MULTIPLAN

Side*Print prints a Multiplan spreadsheet sideways. Included are a documentation file, sample spreadsheet and a disk cataloger/loader program. System requirements are Microsoft Multiplan, Extended BASIC, memory expansion, disk system and a printer that accepts the Star Micronics Gemini 10X protocols for downloading a character set. Send $5, or send a disk, and self-addressed, stamped return mailer to: Jim Swedlow, 7301 Kirby Way, Stanton, CA 90680.

CAPITAL CAPERS

Capital Capers is an educational program designed to help teach when to capitalize a word. All basic rules of capitalization are covered. Students have a choice of being tested only on the basis of a particular rule of capitalization or all rules simultaneously. In both cases, the student presses the space bar or joystick fire button to capitalize a letter that has a cursor over it. The "test" uses jokes and riddles. Students who correctly apply the rules of capitalization in the first part of the joke are rewarded with the punchline. Uses sound effects. Requires Extended BASIC and a cassette recorder. Send cassette and self-addressed, stamped return mailer, or $7, to: Larry P. Morgan, 9 Fountain Lake Dr., Greenfield, IN 46140.

imoed these in your text, use PF PIO, and watch your GEMINI printer do tricks:

```
Enlarged Print          -- CTRL U SHIFT N CTRL U
Cancel Enlarged         -- CTRL U SHIFT T CTRL U
Condensed Print         -- CTRL U SHIFT O CTRL U
Cancel Condensed        -- CTRL U SHIFT R CTRL U
Skip to TOP-OF-PAGE     -- CTRL U SHIFT L CTRL U
Underline Mode          -- CTRL U FCTN R CTRL U SHIFT - CTRL U SHIFT A CTRL U
Cancel Underline        -- CTRL U FCTN R CTRL U SHIFT - CTRL U SHIFT @ CTRL U
Doublestrike Mode       -- CTRL U FCTN R CTRL U SHIFT G
Cancel Doublestrike     -- CTRL U FCTN R CTRL U SHIFT H
Emphasized Mode         -- CTRL U FCTN R CTRL U SHIFT E
Cancel Emphasized       -- CTRL U FCTN R CTRL U SHIFT F
Italic Print            -- CTRL U FCTN R CTRL U 4
Cancel Italic           -- CTRL U FCTN R CTRL U 5
Elite Print             -- CTRL U FCTN R CTRL U SHIFT M
Cancel Elite            -- CTRL U FCTN R CTRL U SHIFT P
Superscript             -- CTRL U FCTN R CTRL U SHIFT S CTRL U SHIFT 2 CTRL U
Subscript               -- CTRL U FCTN R CTRL U SHIFT S CTRL U SHIFT 4 CTRL U
Cancel Super/Sub        -- CTRL U FCTN R CTRL U SHIFT T
6 Lines per inch        -- CTRL U FCTN R CTRL U 2
8 Lines per inch        -- CTRL U FCTN R CTRL U 0
Double-space            -- CTRL U FCTN R CTRL U SHIFT A CTRL U SHIFT T CTRL U
Set skip-over perf.     -- CTRL U FCTN R CTRL U SHIFT N CTRL U SHIFT F CTRL U
Set Left Margin         -- CTRL U FCTN R CTRL U SHIFT M,n
Set Right Margin        -- CTRL U FCTN R CTRL U SHIFT Q,n
```