

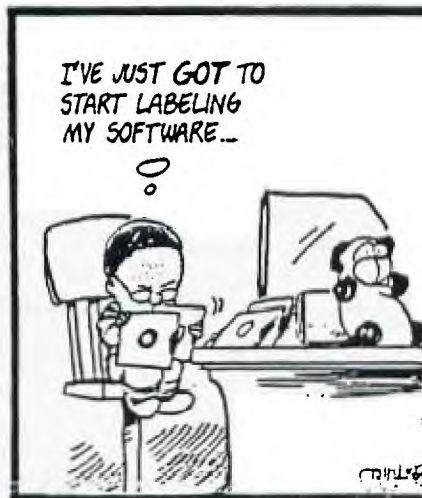
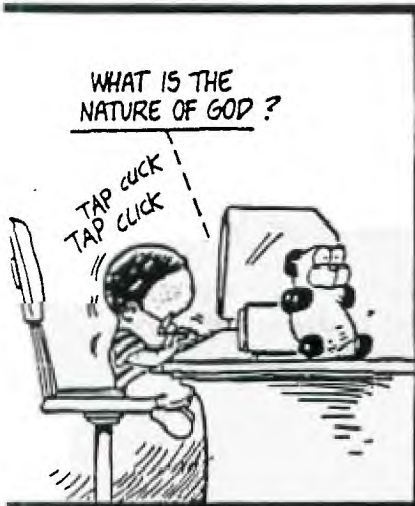


HOCUS

MILWAUKEE
8612021

Home Computer
Users Spotlight
a monthly publication of the
Milwaukee Area 99/4 Users Group

DECEMBER - 1986



MILWAUKEE AREA USER GROUP
4122 GLENWAY WAUWATOSA WI 53222

President...	J.Schroeder	2644735
Vice-Pres...	D.Walden	5292173
Treasurer...	P.Norton	4628954
Secretary...	B.Kling	7742717
Librarian...	E.VonDerEhe	5490593
Librarian...	F.Pabian	3273618
Newsletter..	G.Hitz	5350133
S.I.G.....	Schroeder/Walden/Hitz	

Next Group Meeting
January 10, 1987
Wauwatosa S&L 7500 W. State
1:00PM - 4:00PM

Next S.I.G. Meeting
January 6, 1987
Security S&L 5555 Pt. Wash.
7:00PM - 10:00PM

Annual Membership Dues
Individual - \$10
Family - \$15

Well, January is coming. And that means election of officers again. This month then, we'll be asking for nominations and/or volunteers to fill all the positions. Many of you have just been coming down to our meetings and contributing nothing. Now is the time to pitch in and give us a hand running our organization. One position just begging for any volunteer's the newsletter editor. We gladly accept someone as a co-editor. Spreading the work out'll make it easier for all.

With word processors becoming so common, a child may never have a chance to develop the ineligible hand-writing that could some day qualify him for one of the better paying professions.

ANNOUNCING THE GRAND OPENING OF CSI

Milwaukee's New Full-Service
Computer Center

- Sale and Lease of New and Used Computers
- Used IBM Systems and Peripherals
- New IBM-Compatible Systems and Peripherals
- Extensive Selection of Software and Supplies
- Complete Service and Repair
- Low Prices
- Friendly People

COMPUTER SYSTEMS, INC.

11710 West North Avenue
Wauwatosa, Wisconsin 53226

771-2983

TI WRITER TABLE SORT

by George Paschetto

This program will sort a table written with the TI WRITER. In order to use it, you will need XBASIC.

The program can be very useful for organizing a data base into different indices organized by use, alphabetically, or any other characteristic. Here's an example:

1	2	1	2
4.87	Photo Album, 3 ring	127.72	Camera, Polaroid
29.39	Tripod, silk SM300	39.87	Lens, wide angle
39.87	Lens, wide angle	4.84	Photo Album, 3 ring
127.72	Camera, Polaroid	29.97	Tripod, silk SM300

The first example is sorted on field 1, the price. The second is sorted on field 2, the description. I have used this to compare lists of my students- one sorted by age and the other sorted by reading level. I decided to use files created on the TI WRITER because it supports an 80 character line and it allows easy editing of the table. If your table is part of a larger file, use the ability of the TI WRITER to save or load specific lines of a file.

When creating the table for sorting, you must number the fields of your table by placing the number over the first character in that field. The field marking line must be the first in the file, or the program will not be able to find it.

CORRECT:			INCORRECT:		
1	2	3	1	2	3
2801GP	39.87	Lens, wide angle	2801GP	39.87	Lens, wide angle
300PBX	29.97	Tripod, silk SM300	300PBX	29.97	Tripod, silk
610000PL	127.72	Camera, Polaroid	610000PL	127.72	Camera, Polaroid
3050PEB	4.87	Photo Album, 3 ring	3050PEB	4.84	Photo Album, 3 ring

The incorrect example has misplaced field markers 1 and 2. The 1 (in 127.72) will not be recognized as part of field number 2, and all the entries in field number 1 will be sorted starting with the second character.

Notice that the 2 marking the second field in the correct example is over the left-most digit of the largest number (the 1 in 127.72). Numbers will not be sorted correctly unless they are right-justified. (You may want to use leading zeroes, 009 010 014 240 etc. for ease, but it is not necessary).

In creating your table, plan ahead so that you know what the largest entry will be. It is perfectly alright to have several blank spaces separating fields.

The file can be as long as 400 lines (not counting the field marker) and each line can have up to 9 fields. The fields must all fit on one line.

The sort will always be in ascending order (from smallest to largest).

```

100 REM PRESCAN VARIABLES
110 GOTO 120 :: D$,X,Y,K,S,Z
,FN,FS,FL,L,S$,H,J,B$ :: CAL
L KEY :: CALL SOUND :: CALL
HCHAR
120 CALL CLEAR :: DIM A$(400
),F(10)
130 !@P-
140 INPUT " See instruction
s? Y/N ":D$ :: CALL CLEAR ::
IF D$="N" THEN 230
150 PRINT " This program was
written to sort files create
d with the TI WRITER. The f
ile can be as long as 400 li
nes."
160 PRINT : " The first lin
e of the file must contain t
he field num- bers, like thi
s": :
170 PRINT "1 2 3 <-first
line":"102 NJ PAUL PETERS":
"314 MO CAROL CORRINA":
622 AL HU NOES": :
180 PRINT " You may have up
to 9 fieldsbut they must all
fit on oneline." : : INPU
T " (press enter)":D$ :: CA
LL CLEAR
190 PRINT " The lines can be
up to the full 80 character
s long thatTI WRITER support
s. This program will only
sort one field at a time."
200 PRINT : " The sort is a
lways in ascending orde
r, with the lowest value f
irst."
210 PRINT : " The file can
be used by TI WRITER after t
his program is through wit
hit." : :
220 INPUT " (press enter)":
D$ :: CALL CLEAR
230 DISPLAY AT(12,1):"Input
device and file name,": " D
SK"
240 ACCEPT AT(13,7):D$ :: IF
D$="" THEN 240 ELSE OPEN #1
:"DSK"&D$,DISPLAY ,VARIABLE
80,INPUT
250 CALL CLEAR
260 REM LOOK FOR 1ST LINE
270 LINPUT #1:A$(0):: IF EOF
(1)THEN 600 ELSE IF A$(0)="
" THEN 270
280 FOR X=1 TO 400 :: LINPUT
#1:A$(X):: IF EOF(1)THEN 31
0
290 NEXT X
300 REM F(0)=FIELD'S POSITION
S (COUNT DOWN 2 FROM EOF()-L
AST 2 LINES ARE TABS)
310 X=X-2 :: CLOSE #1 :: FOR
Y=1 TO 9 :: F(Y)=POS(A$(0),
STR$(Y),1):: IF F(Y)=0 THEN
330
320 NEXT Y
330 F(Y)=80 :: Y=Y-1 :: IF Y
THEN 350
340 PRINT "Can't find field
marker.": "This was found i
nstead": " A$(0):: END
350 CALL CLEAR :: DISPLAY AT
(10,5):"Press:": " <1> Sor
t": " <2> Save on disk": "
<3> Quit"
360 CALL KEY(0,K,S):: IF (K<
49)+(K>51)THEN 360 ELSE CALL
SOUND(-20,880,0):: ON K
-48 GOTO 380,540,580

```

--* CONTINUATION OF TI WRITER TABLE SORT *

```

370 REM GET FIELD TO SORT
380 CALL CLEAR :: DISPLAY AT
(1,9):"<choose field>"
390 FOR Z=1 TO Y :: DISPLAY
AT(Z*2,1):"Field#";STR$(Z);"
";SEG$(A$(1),F(Z),F(Z+1)-F(
Z)):: NEXT Z
400 DISPLAY AT(20,1):"Sort o
n field number _." :: FN=0
410 CALL KEY(0,K,S):: IF (K=
13)*(FN>0)THEN 440 ELSE IF (
K<49)+(K>48+Y)THEN 410
420 CALL HCHAR(20,24,K)::FN
=K-48 :: GOTO 410

430 REM SORTING (FN=FIELD NI
MBER, FS=FIELD'S START, FL=F
IELD'S LENGTH)
440 CALL CLEAR :: DISPLAY AT
(12,5):"Sorting..." :: FS=F(
FN):: FL=F(FN+1)-FS :: K=X :
: L=INT(X/2)+1
450 IF L<>1 THEN L=L-1 :: S$
=A$(L):: GOTO 470
460 S$=A$(K):: A$(K)=A$(1)::
K=K-1 :: IF K<1 THEN A$(H)=
S$ :: GOTO 350
470 J=L
480 H=J :: J=J+J :: IF J>K T
HEN A$(H)=S$ :: GOTO 450

490 IF J)=K THEN 510
500 IF SEG$(A$(J),FS,FL)<SEG
$(A$(J+1),FS,FL)THEN J=J+1
510 IF SEG$(S$,FS,FL)>SEG$(
A$(J),FS,FL)THEN A$(H)=S$ ::
GOTO 450
520 A$(H)=A$(J):: GOTO 480
530 REM SAVE ON DISK (Y GOES
TO X+2 SO LAST 2 LINES OF F
ILE ARE SAVED)
540 CALL CLEAR :: DISPLAY AT
(12,1):"Input device and fil
e name,": " DSK";D$
550 ACCEPT AT(13,7)SIZE(-12)
:D$ :: IF D$="" THEN 240 ELS
E OPEN #1:"DSK";D$,DISPLAY
VARIABLE 80,OUTPUT
560 FOR Y=0 TO X+2 :: PRINT
#1:A$(Y):: NEXT Y :: CLOSE
1 :: GOTO 350

570 REM QUIT
580 CALL CLEAR :: DISPLAY A
(12,1):"QUIT- Are you sure?
Y/N"
590 ACCEPT AT(12,25)SIZE(1)
:ALIDATE("YN"):B$ :: IF B$="
" THEN 350 ELSE END
600 PRINT "File not properl
y organized." :: GOTO 230

```

Memory Magic
by Ken Schmidt

If any of you have ever loaded Extended Basic Object code with the statement CALL LOAD "DSKx.FileName", you know it don't load very fast. A solution to this problem is to use the Utility "SYSTEX" which saves the Xbasic Object code together with a loader into a Program Image file which will load fast with the command RUN "DSKx.FileName".

I used the "SYSTEX" utility to save the object code of a program that redefines the keyboard so that CTRL keys 0-9 & A-Z can be used to input strings up to 24 characters to be used in writing programs or command phrases.

Then I decided to try the usefulness of the program which is called "KEYLOAD". Started typing in a long basic program using the new key program. Saves a lot of typing on new programs. Then saved it to disk and ran the catalog to make sure it was saved O.K. Much to my surprise the files size was over 30 sectors when I expected it to be about 5 or 6. Upon listing the program I found the visible lines of the "SYSTEX" program at the beging of the program I had just typed in. Using the SIZE command I found that there was over

10,000 bytes of program space being used. It then dawned on me that I did not typed new before starting.

I was determined not retype all the code over so I started experimenting. Delete "SYSTEX" visible lines and CALL INIT, resave. No luck because "SYSTEX" has a line #0 which cannot be deleted. Resequece and delete lines. Did not work because deleting the line that was #0 caused the computer to lock up.

Here is the solution which worked great. First delete all visible "SYSTEX" lines except #0. List program to disk, LIST "DSKx.FileName". Load this file in TI-Writer and delete line #0. Resave file. Use a transfer program such as "XLATE" or "CONVRT" to change back to a program. This is in Dis/Var 163 format. Type "NEW" and load D/V 163 file with MERGE "DSKx.FileName". Then resave as a program.

If any of you ever have this kind of experince you now know how to fix the error without retyping eveything.

The reference to visible lines above means that when object code is saved with "SYSTEX" you have 10 lines of Extended Basic that must not be deleted or the "SYSTEX"ed file will not work.

```

100 REM ----- CAD -----
- COMPUTER AIDED DESIGN JULY
  1, 1985, S. L. JOHNSTON
110 OPEN #1:"PI0"
120 DIM GRAPHIC(160)
130 CALL CLEAR
140 PRINT "SELECT COLORS"
150 PRINT
160 PRINT "CHOOSE COLOR CODE
FROM 1 TO 16"
170 PRINT
180 INPUT "SCREEN COLOR? ":
S
190 PRINT
200 INPUT "CHARACTER COLOR?
":F
210 REM -----
----- GET SCR
EEN AND CHARACTER COLORS
220 CALL SCREEN(S)
230 FOR C=1 TO 16
240 CALL COLOR(C,F,S)
250 NEXT C
260 CALL CLEAR
270 REM -----
----- DETERMIN
E HARDWARE CONFIGURATION
280 PRINT "HARDWARE CONFIGUR
ATION"
290 PRINT
300 PRINT "ENTER YES OR NO"
310 PRINT
320 INPUT "DO YOU HAVE A DIS
K DRIVE? ":DRIVE#
330 PRINT
340 IF DRIVE#="YES" THEN 360
350 IF DRIVE#="NO" THEN 410
360 INPUT "FILENAME? (DISK1:
name) ":LABEL#
370 OPEN #2:LABEL#,INTERNAL,
FIXED 21#
380 PRINT
390 INPUT "LOADING DESIGN FR
OM DISK? ":START#
400 PRINT
410 INPUT "DO YOU HAVE A GEM
INI PRINTER? ":PRINTER#
420 CALL CLEAR
430 REM -----
----- DEFINE ALPHANUMER
IC SYMBOLS FOR HARD COPY
440 FOR J=1 TO 160
450 GRAPHIC(J)=J
460 NEXT J
470 REM -----
----- REDEFINE BLOCK SHAP
IC SYMBOLS FOR HARD COPY
480 GRAPHIC(129)=237
490 GRAPHIC(132)=238
500 GRAPHIC(134)=239
510 GRAPHIC(135)=240
520 GRAPHIC(136)=246
530 GRAPHIC(138)=245
540 GRAPHIC(139)=247
550 GRAPHIC(140)=244
560 GRAPHIC(147)=232
570 GRAPHIC(155)=249
580 GRAPHIC(133)=236
590 GRAPHIC(137)=242
600 GRAPHIC(143)=243
610 GRAPHIC(144)=248
620 GRAPHIC(145)=235
630 GRAPHIC(146)=233
640 GRAPHIC(148)=234
650 GRAPHIC(149)=241
660 GRAPHIC(151)=231

```

```

670 GRAPHIC(153)=240
680 REM -----
----- D
EFINE GRAPHIC CHARACTERS
690 CALL CHAR(151,"FFFFFFFFF
0000000")
700 CALL CHAR(147,"0000000FF
FFFFFFF")
710 CALL CHAR(146,"0F0F0F0F
0F0F0F0")
720 CALL CHAR(148,"0F0F0F0F
0F0F0F0")
730 CALL CHAR(145,"FFFFFFF
0F0F0F0")
740 CALL CHAR(133,"FFFFFFF
0F0F0F0")
750 CALL CHAR(129,"F0F0F0F
FFFFFF")
760 CALL CHAR(132,"0F0F0F0
FFFFFF")
770 CALL CHAR(134,"FFFFFFF
FFFFFF")
780 CALL CHAR(153,"000000F
080808")
790 CALL CHAR(149,"000000F
0000000")
800 CALL CHAR(137,"000000F
080808")
810 CALL CHAR(143,"000000F
080808")
820 CALL CHAR(140,"080808F
080808")
830 CALL CHAR(138,"080808
080808")
840 CALL CHAR(136,"080808F
0000000")
850 CALL CHAR(139,"080808F
0000000")
860 CALL CHAR(144,"080808F
0000000")
870 CALL CHAR(156,"080808F
080808")
880 CALL CHAR(135,"080808F
080808")
890 REM -----
----- LOAD PREVIOUSLY DEV
ELOPED DESIGN FROM DISK
900 IF DRIVE#="NO" THEN 1000
910 IF START#="YES" THEN 930
920 IF START#="NO" THEN 1000
930 FOR COL=1 TO 32
940 FOR ROW=1 TO 24
950 INPUT #2:A,
960 CALL HCHAR(ROW,COL,A,1)
970 NEXT ROW
980 NEXT COL
990 REM -----
----- DISPLAY DESIGN
1000 R=12
1010 C=1#
1020 CURSOR=43
1030 CALL KEY(C,KEY,STATUS)
1040 CALL GCHAR(R,C,A)
1050 CALL HCHAR(R,C,CURSOR,1)
1060 CALL HCHAR(R,C,A,1)
1070 IF STATUS=0 THEN 1030
1080 REM -----
----- CHECK FOR ARROW KEY
1090 IF KEY<>11 THEN 1140
1100 R=R-1
1110 IF R<0 THEN 1030 ELSE

```

```

1120 R=1
1130 GOTO 1030
1140 IF KEY<>10 THEN 1190
1150 R=R+1
1160 IF R<>25 THEN 1030 ELSE
1170
1170 R=24
1180 GOTO 1030
1190 IF KEY<>8 THEN 1240
1200 C=C-1
1210 IF C<>0 THEN 1030 ELSE
1220
1220 C=1
1230 GOTO 1030
1240 IF KEY<>9 THEN 1300
1250 C=C+1
1260 IF C<>33 THEN 1030 ELSE
1270
1270 C=32
1280 GOTO 1030
1290 REM -----
----- CHECK
FOR SAVE, PRINT OR BOTH
1300 IF KEY=1 THEN 1370
1310 IF PRINTER#="NO" THEN 1
340
1320 IF KEY=6 THEN 1510
1330 IF KEY=15 THEN 1370
1340 CALL HCHAR(R,C,KEY,1)
1350 GOTO 1030
1360 REM -----
----- SAVE DESIGN ON DISK
1370 IF DRIVE#="NO" THEN 1340
1380 RESTORE #2
1390 FOR COL=1 TO 32
1400 FOR ROW=1 TO 24
1410 CALL GCHAR(ROW,COL,A)
1420 PRINT #2:A,
1430 NEXT ROW
1440 PRINT #2:
1450 NEXT COL
1460 CLOSE #2
1470 IF KEY=15 THEN 1510
1480 END
1490 REM -----
----- PRINT DESIGN ON GEMINI
1500 REM -----
12 LINES/INCH, UNIDIRECTIONA
L MODE, EMPHASIZED MODE
1510 PRINT #1:CHR$(27);CHR$(
55);CHR$(6);CHR$(27);CHR$(85)
);CHR$(1);CHR$(27);CHR$(59);
1520 FOR ROW=1 TO 24
1530 FOR COL=1 TO 32
1540 CALL GCHAR(ROW,COL,A)
1550 PRINT #1:CHR$(GRAPHIC(C
));
1560 NEXT COL
1570 PRINT #1
1580 NEXT ROW
1590 REM -----
----- RESET PRINT
ER TO POWER ON CONDITION
1600 PRINT #1:CHR$(27);CHR$(
54);
1610 REM -----
----- PRINT FILENAM
E DESIGN IS STORED UNDER
1620 FOR I=1 TO 5
1630 PRINT #1
1640 NEXT I
1650 IF DRIVE#="NO" THEN 167
0 ELSE 1660
1660 PRINT #1:"STORED UNDER
FILENAME ":LABEL#
1670 END

```

```

100 PRINTSTRIP BY:
    MICHAEL A. MACHONIS /
    SEVERNA PARK, MD /
110 /
    DELETE LINE #510 IF /
    YOUR PRINTER DOES /
    NOT SUPPORT SUPER- /
    SCRIPT. /
120 /
130 CALL CLEAR :: CALL SCREE
N(1)
140 CALL CHAR(100,"000000000
000FFF")
150 CALL CHAR(104,"FFFF")
160 CALL CHAR(101,"030303030
3030303")
170 CALL CHAR(102,"303030303
3030303")
180 DISPLAY AT(9,1):RPT$(4"
.20:") e PRINT-A-STRIPe
e":RPT$(4"b.20"):RPT$(4"
.20:") eBY MIKE MACHONIS
e e":RPT$(4"b.20")
185 CALL SOUND(150,1397,0)
190 DISPLAY AT(24,3):"PRESS
ANY KEY TO CONTINUE" :: CALL
KEY(O,K,S):: IF S=0 THEN 19
0
200 CALL CHARSET
210 DIM A(11),B$(2,11),C$(2,
11),E$(13),G$(99)
220 P$(2)=CHR$(27)"["CHR$(
27)"E"CHR$(27)"G"CHR$(27
)"-]" :: Q(2)=80 :: N$(2)="
#####"
230 P$(1)=CHR$(27)"["CHR$(
15)CHR$(27)"G"CHR$(27)"-
]" :: U(1)=136 :: W$(1)="###
#####"
235 IN=1 :: P=1 :: S=1 :: CH
=1
240 OPEN #1:"PI0.CR"
250 K=0 :: C$="UPPER (CTRL)"
260 DISPLAY AT(12,1)ERASE AL
L:"> INPUT FROM KEYBOARD":
"> INPUT FROM DISK (DSK1.)
": "YOUR CHOICE": "IN :: AC
CEPT AT(16,15)BEEP SIZE(-1)V
ALIDATE("12"):IN :: IF IN=1
THEN 290
270 GO SUB 620
280 OPEN #3:"DSK1."JF$ :: IN
PUT #3:P :: FOR I=1 TO 2 ::
FOR J=1 TO 11 :: LINPUT #3:A
$(I,J):: NEXT J :: NEXT I ::
CLOSE #3 :: GO TO 370
290 DISPLAY AT(12,1)ERASE AL
L:"> COMPRESSED PRINT": ">
NORMAL PRINT": "YOUR CHOI
CE": "P :: ACCEPT AT(16,15)B
EEP SIZE(-1)VALIDATE("12"):P
300 IF P=1 THEN RESTORE 800
ELSE RESTORE 810
310 FOR I=1 TO 11 :: READ A(
I):: NEXT I
320 K=K+1
330 DISPLAY AT(1,4)ERASE ALL
:"INPUT "C$) ROM"
340 FOR I=2 TO 18 STEP 2 ::
DISPLAY AT(I+2,1):"OVER KEY"
;I/2 :: NEXT I :: DISPLAY AT
(22,1):"OVER KEY 0": "OVER
KEY ="
350 FOR I=2 TO 22 STEP 2 ::
J=I/2 :: ACCEPT AT(I+2,12)B
EEP SIZE(A(J)):A$(K,J):: NEXT
I

```

```

355 GO SUB 870 :: IF Y$="N"
OR Y$="n" THEN 330
360 IF K=1 THEN C$="LOWER (F
CTN)" :: GO TO 320
370 DISPLAY AT(12,1)ERASE AL
L:"HOW MANY STRIPS: "S :: A
CCEPT AT(12,19)BEEP SIZE(-4)
VALIDATE(DIGIT):S
380 IF P=1 THEN RESTORE 800
ELSE RESTORE 810
390 FOR I=1 TO 11 :: READ A(
I):: E$(A(I))=" "RPT$(#",A
(I)):: NEXT I
400 PRINT #1:P$(P)
410 FOR C=1 TO 8
420 PRINT #1:RPT$(#",Q(P));
CHR$(13):CHR$(10)
430 K=1
440 FOR I=1 TO 11
450 IF LEN(A$(K,I))<A(I)THEN
B$(K,I)=RPT$(#",A(I))-LEN(
A$(K,I))/2)A$(K,I)ELSE B$(
K,I)=A$(K,I)
460 IF I>1 THEN 470 ELSE PR
INT #1,USING W$(P):B$(K,I)::
GO TO 480
470 PRINT #1,USING E$(A(I)):
B$(K,I)
480 NEXT I
490 PRINT #1:CHR$(13)CHR$(1
0)
500 K=K+1 :: IF K=2 THEN 440
510 PRINT #1:CHR$(27)"-0":C
HR$(27)"J"CHR$(1):CHR$(27
)"50":RPT$(#",Q(P)):CHR$(27
)"T":CHR$(27)"-]"
520 PRINT #1:RPT$(CHR$(10),2
)
530 NEXT C
540 PRINT #1:CHR$(27)"["
550 IF IN=2 THEN 600
560 DISPLAY AT(12,1)ERASE AL
L:"SAVE TO DISK? Y/N"
570 CALL KEY(O,K,S):: IF S=0
THEN 570 ELSE IF (K=89)OR(K
=121)THEN 580 ELSE 600
580 DISPLAY AT(12,1)ERASE AL
L:"ENTER FILE NAME": "DSK
1." :: ACCEPT AT(15,5)BEEP S
IZE(9):F$ :: F$="DSK1."F$)
590 ON ERROR 830 :: OPEN #2:
F$ :: PRINT #2:P :: FOR I=1
TO 2 :: FOR J=1 TO 11 :: PRI
NT #2:A$(I,J):: NEXT J :: NE
XT I :: CLOSE #2
600 DISPLAY AT(12,1)ERASE AL
L:"QUIT PROGRAM? (Y/N)": "Y
OUR CHOICE: N" :: ACCEPT AT
(14,15)SIZE(-1)BEEP VALIDATE
("Ynvn"):Y$
610 IF (Y$="N")OR(Y$="n")THE
N 250 ELSE 820
620 ON ERROR 850 :: OPEN #4:
"DSK1.",INPUT ,RELATIVE,INTE
RNAL
630 FOR L=1 TO 127
640 INPUT #4:FN$ :: IF FN$="
" THEN 670
650 IF SEG$(FN$,LEN(FN$),1)=
"." THEN F=F+1 :: G$(F)=FN$
660 NEXT L
670 CLOSE #4
680 CALL CLEAR
681 IF F>1 THEN 690
682 DISPLAY AT(12,2):"NO FIL
ES FOUND, TRY AGAIN" :: FOR
C=1 TO 200 :: NEXT C :: GO T
O 250
690 FOR N=1 TO F

```

```

700 DISPLAY AT(2,2):USIN
E "#) #####":M.SEG$(G$(
N),1,LEN(G$(N))-1)
710 Z=Z+1 :: IF Z=11 THEN Z=
0 :: CK=CK+1
720 IF CK/2=INT(CK/2)THEN N=
1 ELSE N=16
730 IF (INT(N/22)=N/2)THEN 7
40 ELSE 750
740 DISPLAY AT(24,1):"DISPLA
Y MORE FILES? (Y/N)" :: CALL
KEY(O,K,S):: IF S=0 THEN 74
0
750 IF (K=89)OR(K=121)THEN C
ALL CLEAR ELSE 770
760 NEXT M
770 DISPLAY AT(24,1):"YOUR C
HOICE: "CH :: ACCEPT AT(24,
15)BEEP SIZE(-2)VALIDATE(DIG
IT):CH :: IF (CH=0)OR(CH#F)T
HEN 770
780 F$=G$(CH): F,CK,Z,N=0
790 RETURN
800 DATA 1,1,1,1,1,1,1,1,1
1,1,1,1,1,1,1,1,1,1,1,1,1
810 DATA 6,6,7,6,7,6,6,7,7,6
,6
820 CALL INIT :: CALL PEEK(2
,J,K):: CALL LOAD(-31804,J,K
):: STOP
830 CALL SCREEN(7):: DISPLAY
AT(20,1):"NO DISK IN DRIVE"
:"PRESS ANY KEY TO RETRY" ::
GO TO 860
850 CALL SCREEN(7):: DISPLAY
AT(20,1):"NO DISK/FILE IN D
RIVE": "PRESS ANY KEY TO RETR
Y"
860 CALL ERR(E1,E2,E3,E4)::
CALL SOUND(150,1397,0)
865 IF E4=620 OR E4=640 THEN
PRINT : "WHEN DISC IS PLAC
ED IN "DRIVE #1, PRESS ANY
KEY":
866 CALL KEY(O,K,S):: IF S=0
THEN 866 :: CALL CLEAR :: C
ALL SCREEN(11)
867 IF E4=620 OR E4=640 THEN
RUN
869 ON ERROR 830 :: RETURN 8
00
870 DISPLAY AT(2,2):"ENTRIES
CORRECT? (Y/N) Y" :: ACCEPT
AT(2,25)SIZE(-1)VALIDATE("Y
vnv")BEEP:Y$ :: RETURN

```

I. I. CAD

FCIN - ARROW KEYS TO MOVE
FCIN 7- SAVE TO DISK
FCIN 8- PRINT(GEMINI)
FCIN 9- SAVE AND PRINT

* Computer will ignore FCIN
printer or disk drive.

CTRL- P W E R T Y U I O P
CTRL- A S D F G H J K L ; ' -

Copyright 1986

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 138 original programs in Basic and Extended Basic, available on cassette or disk, only \$3.88 each plus \$1.58 per order for PPM. Entertainment, education, programmer's utilities. Descriptive catalog \$1.88, deductible from your first order.

Tips from The Tigercub, a full disk containing the complete contents of this newsletter Nos. 1 through 14, 58 original programs and files, just \$15 postpaid.

Tips from the Tigercub Vol. 2, another diskfull, complete contents of Nos. 15 through 24, over 68 files and programs, also just \$15 postpaid.

Tips from the Tigercub #
Vol. 3 is now ready. #
Another 62 programs, #
routines, tips, tricks. #
from Nos. 25 thru 32. #
Also \$15 postpaid. Any #
two Tips disks \$27 or #
all 3 for \$35 postpaid. #

Nuts & Bolts (No. 1), a full disk of 188 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 188 utility subprograms in merge format, all new and fully compatible with the last, and with 18 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 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, send a dollar for my catalog! The READFILE subprogram on my Nuts & Bolts #2 disk has a backward parentheses in line 21161. This is the corrected line -
21161 DISPLAY AT(17,1):"OPEN PRINTER #:"NAME? " : ACCE
PT AT(17,15)VALIDATE(DIGIT)S
IZE(-3):P : ACCEPT AT(18,7)
:P\$: OPEN #P:P\$: GOTO 21
163

When Texas Instruments developed Extended Basic, they took away the ability of Basic to redefine or color the characters in sets

15 and 16, ASCII 144 to 159, in order to make room in memory for sprites (they did let us have color set # instead. That is why Basic programs which use sets 15 and 16 will crash if you try to run them in XBasic.

Finally, John Behnke published in the Chicago Times newsletter an amazing routine which gave us back those missing sets. His routine was 13 sectors long. Recently, Richard Heath published in the L.A. newsletter a shortened version. And, without having any idea how it works, I have managed to scrunch it down to only 4 sectors -

```
1 CALL BXB
29999 !BXB by Jim Peterson,
adapted from VDPUTIL2 by Joh
n Behnke/Richard Heath
38888 SUB BXB :: CALL INIT :
: CALL LOAD(8194,37,194,63,2
48)
38881 CALL LOAD(16368,88,79,
67,72,65,82,37,58,88,79,75,6
9,86,32,37,168)
38882 !
38883 FOR J=1 TO 136 :: CALL
LOAD(9529+J,ASC(SEG$(J)\[J$
,J,1))): NEXT J :: SUBEND
38884 SUB CHAR(A,A$):: CALL
LOAD(9588,A):: CALL LINK("PO
CHAR",A$):: SUBEND
38885 SUB COLOR(A,B,C):: CAL
L LOAD(9492,8,15+A,(8-1)*16+
C-1)
38886 CALL LINK("POKEV"):: S
UBEND
```

Note than line 38882 is missing. That's because there is no way to key it in. Once again we need a program that writes a program -

```
188 FOR J=1 TO 136 :: READ A
:: M$=M$&CHR$(A):: NEXT J
118 OPEN #1:"DSK1.BXBADATA",V
ARIABLE 163,OUTPUT :: PRINT
#1:CHR$(125)&CHR$(8)&"[I[$
"&CHR$(198)&CHR$(199)&CHR$(1
36)&M$&CHR$(8)
128 PRINT #1:CHR$(255)&CHR$(
255):: CLOSE #1
138 DATA 2,224,37,28,3,8,8,8
```

```
,2,5,48,48,2,6,37,2,285,133,
2,134,37,17
148 DATA 17,252,4,192,2,1,8,
1,2,2,37,1,2,3,18,8,212,131,
4,32,32,28
158 DATA 288,4,9,88,2,32,3,8
,2,1,37,2,2,2,8,8,2,7,11,8,2
,8,7,8,193
168 DATA 1,192,193,193,188,9
7,133,145,135,21,1,113,136,6
,198,145
178 DATA 135,21,1,113,136,21
8,78,18,198,177,137,228,198,
2,131,37,18
188 DATA 17,248,4,32,32,36,1
6,6,2,224,37,28,3,8,8,8,4,32
,32,32,4
198 DATA 192,216,8,131,124,2
,224,131,224,4,96,8,112
```

RUN that to create a file BXBADATA on the disk. Then load the BXB program, and enter MERGE DSK1.BXBADATA. The unprintable line will pop into place. SAVE this completed BXB routine in MERGE format, and merge it into any Basic-only program. If you want, the result can be run through a Compactor program and turned into multi-statement program lines for more speed.

Or, you can write an Extended Basic program using all 16 character sets for graphics and color - actually 17, because set # is also available. Even the characters 24 through 31 can be redefined! Craig Miller has warned against fooling around in that area of memory, but there seems to be no problem with redefining the cursor (38) or the edge character (31).

Sprites can only use characters between 32 and 143 and their color cannot be changed with CALL COLOR(#, _). I have not found any other bugs, but have not had time for much experimenting.

Here's an easy Tigercub challenge - run this one in Basic, not Extended Basic.

```
>LIST
100 DISPLAY AT(1,1):#
>RUN
#
#
Why did it print the zero
twice?
```

I wrote this next one primarily for blind users. It converts each PRINT or DISPLAY directly to speech output and also provides a speech prompt for INPUTs.

```
100 !PRINT SPEAKER by Jim Peterson - to add OPEN #1:"SPEECH",OUTPUT and convert PRINT and DISPLAY statements to PRINT #1
110 !Also writes a PRINT #1 for INPUT prompts
120 !Program to be converted must first be SAVED in MERGE format. Recommend it be RESequenced before SAVEing, to make room for INPUT lines
130 PS=CHR$(156)&CHR$(253)&CHR$(200)&CHR$(1)&"1"&CHR$(181)
140 DISPLAY AT(3,1)ERASE ALL:"INPUT FILENAME?":"DSK"::ACCEPT AT(4,4):IF$::OPEN #1:"DSK"&IF$,INPUT ,VARIABLE 163
150 DISPLAY AT(5,1):"OUTPUT FILENAME?":"DSK"::ACCEPT AT(6,4):OF$::OPEN #2:"DSK"&OF$,OUTPUT,VARIABLE 163
160 PRINT #2:CHR$(0)&CHR$(1)&CHR$(159)&CHR$(253)&CHR$(200)&CHR$(1)&"1"&CHR$(181)&CHR$(199)&CHR$(16)&"SPEECH"&CHR$(179)&CHR$(247)&CHR$(0)
170 LINPUT #1:M$::P=POS(M$,CHR$(156),3)::A=POS(M$,CHR$(162),3)::Z=POS(M$,CHR$(181),3)
180 I=POS(M$,CHR$(146),1)::IF I=# THEN 210::IF Z=# OR Z<I THEN PRINT #2:M$::GOTO 240
190 M2$=SEG$(M$,1,1)&SEG$(M$,2,1)&PS&SEG$(M$,I+1,Z-I-1)&CHR$(0)::PRINT #2:M2$
200 PRINT #2:SEG$(M$,1,1)&CHR$(ASC(SEG$(M$,2,1))+1)&SEG$(M$,3,255)::GOTO 240
210 IF P+A=# THEN PRINT #2:M$::GOTO 240
```

```
220 M=MAX(P,A)
230 M$=SEG$(M$,1,2)&PS&SEG$(M$,M+1,255)::PRINT #2:M$
240 IF EOF(1)<>1 THEN 170 ELSE CLOSE #1::CLOSE #2
250 DISPLAY AT(12,1)ERASE ALL:"Type NEW and Enter"::DISPLAY AT(15,1):"Type MERGE DSK":OF$::END
*****
MOLLY DARLING
100 CALL CLEAR::CALL SCREEN(5)::FOR SE=1 TO 12::CALL COLOR(SE,16,5)::NEXT SE
110 DISPLAY AT(3,8):"MOLLY DARLING"::"Written and performed by":TAB(9):"Eddy Arnold"::DISPLAY AT(24,1):"Programmed by Jim Peterson"
120 FOR D=1 TO 200::NEXT D::DISPLAY AT(12,1):"Just a moment....."::".....looking for my music..."
130 DIM N(100),N2(100),A(250),B(250),C(250)::F=110::FOR J=1 TO 80::N(J)=INT(F=.859463894^(J-1)+.5)::NEXT J
140 DATA 16,11,8,16,8,11,16,4,11,18,11,8
150 DATA 20,16,11,23,11,16,25,21,16,28,16,21
160 DATA 23,20,16,23,16,20,23,11,16,23,16,11
170 DATA 20,11,16,20,16,11,20,8,11,20,11,8
180 DATA 20,11,16,25,16,11,23,11,16,20,8,4
190 DATA 18,16,10,18,10,16,18,16,10,18,11,16
200 DATA 18,15,11,18,9,15,18,11,9,18,9,3
210 DATA 20,8,1,20,13,0,20,8,13,20,13,4
220 DATA 27,20,18,27,18,20,20,18,12,20,12,18
230 DATA 25,21,16,25,16,21,25,13,16,25,16,13
240 DATA 27,23,21,27,21,23,27,23,18,27,18,21
250 DATA 28,23,20,28,20,23,28,20,16,27,16,20
260 DATA 30,21,13,28,13,21,27,21,13,25,13,21
270 DATA 23,20,16,23,16,20,20,11,16,20,16,11
280 DATA 30,23,13,28,13,23,23,20,13,20,13,16
290 DATA 25,21,16,25,16,21,25,21,16,27,16,21
```

```
300 DATA 28,23,20,20,16,11,18,15,11,20,11,15
310 DATA 16,11,8,16,8,11,16,9,1,16,1,9
320 DATA 16,11,8,16,8,11,16,1,8,16,13,1
330 DATA 25,21,16,25,16,13,25,13,9,25,9,4
340 DATA 23,20,16,23,16,11,23,11,8,23,0,4
350 DATA 21,18,11,21,11,9,21,9,6,20,6,3
360 DATA 21,16,11,20,16,11,20,11,8,20,8,4
370 DATA 18,13,10,18,10,6,18,6,1,20,13,10
380 DATA 22,18,13,28,22,18,27,18,22,25,22,18
390 DATA 23,18,15,23,15,11,23,11,6,23,6,3
400 DATA 23,21,15,23,15,11,23,11,9,23,9,6
410 DATA 16,13,8,16,8,13,16,13,8,18,13,9
420 DATA 20,11,8,21,8,11,20,11,8,18,11,6
430 RESTORE 140::T=16::GOSUB 400::RESTORE 140::T=4::GOSUB 400::RESTORE 140::T=12::GOSUB 400::RESTORE 140::T=16::GOSUB 400
440 RESTORE 210::T=20::GOSUB 400::RESTORE 170::T=4::GOSUB 400::RESTORE 250::T=4::GOSUB 400::RESTORE 200::T=8::GOSUB 400::RESTORE 140::T=16::GOSUB 400::RESTORE 290::T=40::GOSUB 400::RESTORE 140::T=16::GOSUB 400::RESTORE 410::T=8::GOSUB 400
460 RESTORE 310::T=8::GOSUB 400::GOTO 490
470 GOTO 490
480 FOR J=1 TO T::X=X+1::READ A(X),B(X),C(X)::A(X)=A(X)+12::B(X)=B(X)+12::C(X)=C(X)+12::NEXT J::RETURN
490 DISPLAY AT(10,1):"Control volume of 3 voices":"using 1, 2 and 3 keys for":"louder and Q, W and E for":"softer."::"
500 DISPLAY AT(15,1):"Control speed using 'F' for":"faster and 'S' for slower."
```

```
510 DISPLAY AT(18,1):"Change key using 'A' for":"higher and 'D' for lower."
520 DISPLAY AT(21,1):"Press 'Z' for minor key, 'X' for major key."::V1,V2,V3=10::F,P,Y=#::X=200
530 FOR J=1 TO 192::CALL SOUND(-999,N(A(J)-Y),V1,N(B(J)-Y),V2,N(C(J)-Y),V3)::FOR T=1 TO X/50::P=1^X::NEXT T
540 CALL KEY(0,K,S)::IF S<1 THEN 710::ON POS("123QWERTYUIOPASDFGHJKLZXCVBNM"),CHR$(K),1)+1 GOTO 710,550,560,570,580,590,600,610,620,630,650,670,690
550 V1=V1-1-(V1=0)::GOTO 710
560 V2=V2-2-(V2=0)#2::GOTO 710
570 V3=V3-2-(V3=0)#2::GOTO 710
580 V1=V1+2+(V1=30)#2::GOTO 710
590 V2=V2+2+(V2=30)#2::GOTO 710
600 V3=V3+2+(V3=30)#2::GOTO 710
610 X=X-20-(X<2)#20::GOTO 710
620 X=X+20::GOTO 710
630 IF F=1 THEN GOSUB 700
640 Y=Y-1-(Y=-20)::GOTO 710
650 IF F=1 THEN GOSUB 700
660 Y=Y+1+(Y=6)::GOTO 710
670 IF F=1 THEN 710::GOSUB 680::GOTO 710
680 F=1::Y=0::FOR W=3 TO 27 STEP 12::N2(W)=N(W)::N(W)=N(W-1)::N2(W+5)=N(W+5)::N(W+5)=N(W+4)::N2(W+10)=N(W+10)::N(W+10)=N(W+9)::NEXT W::RETURN
690 IF F=0 THEN 710::GOSUB 700::GOTO 710
700 F=0::FOR W=3 TO 27 STEP 12::N(W)=N2(W)::N(W+5)=N2(W+5)::N(W+10)=N2(W+10)::NEXT W::RETURN
710 NEXT J::J=192::FOR V=10 TO 30::CALL SOUND(-999,N(A(J)-Y),V,N(B(J)-Y),V,N(C(J)-Y),V)::NEXT V::FOR D=1 TO 500::NEXT D::GOTO 530
```

MEMORY FULL
Jim Peterson

MILWAUKEE AREA 99-4 USER GROUP
4122 N. GLENWAY
WAUKATOSA, WI 53222



Edmonton 99 UG

box 11983

Alberta

Canada

T5J 3L1

