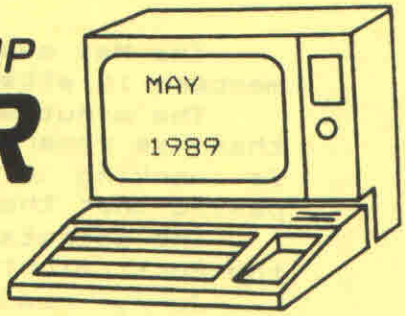
CEDAR VALLEY 99'ER USER GROUP NEWSLETTER



CEDAR RAPIDS/MARION, IOWA

****

OFFICERS

PRESIDENT:
Jerry Canady
6616 Kent Dr. NE
Cedar Rapids Iowa 52402
(319) 377-9382 (Home) or
(319) 395-2494 (Office)

VICE PRESIDENT:
Gary Bishop
3270 28th Ave.
Marion, Iowa 52302
(319) 377-9574

SECRETARY:
Bill Paeth
923 Owen St. NW
Cedar Rapids, Iowa 52405
(319) 396- 6470

TREASURER:
Bruce Winter
702 Fernwood Dr. NE
Cedar Rapids, IA 52402
(319) 393-0610

COMMITTEES

PROGRAM:
Ed Edwards
102 N. Davis St.
Anamosa, Iowa 52205
462-2329

PUBLICITY:
Paul Mortensen
301 Pebble Lane
Hiawatha, Iowa 52233
393-6022

EDUCATION/LIBR ARIAN:
John Johnson
398 Forest Dr. SE
Cedar Rapids, Iowa 52403
366-4541

EDITOR:
Jim Green
288 Windsor Dr. NE
Cedar Rapids, Iowa 52402
377-4073 (Home) or
395-1898 (Office)

****NEWSLETTER TOPICS****

1. Future Meeting Dates
2. Next Meeting Notes
3. Minutes from the May Meeting
4. Vice Squawk
5. File Conversion Program
6. Maxisprites
7. Tips from the Tigercub #52
8. Sector Sharing

****FUTURE MEETING DATES****

Please mark the following dates on your calendar for future meetings: JUNE 12, JULY 10, AUGUST 14.

********NEXT MEETING********

The regular monthly meeting will be Monday, June 12, at West Music, Cedar Rapids. Opening is at 6:30 PM. The program for the evening will be a surprise; come and see what Ed has planned for us!

* MINUTES FROM THE MAY MEETING *

The May meeting was called to order by President Jerry Canady with 12 members in attendance. Also in attendance were two visitors.

The minutes of the April meeting were discussed. Jim Reiss requested that his remarks about the new 80 column card be changed to indicate he is working with Tony Lewis on this project. It was moved, seconded and passed that the printed minutes as amended above be approved.

The secretary wishes to thank Bob Wahlstrom for taking and writing the April minutes on such short notice.

Jerry read the treasurer's report that Bruce left with him. It was not a formal treasury report but indicated that we were still in the black. As in past months, no bank statement had been received by meeting day.

OLD BUSINESS: 1. Jerry announced that he had finally got DM 1000 and FUNNELWEB working again. Being the "ultimate tinkerer" as he called himself, he found that he had made one too many changes in the copy of these programs that he was using. After recopying from the master disk in his file everything worked properly. 2. John Johhson asked that we formally accept the CATCOM program we are using on the library. It was moved, seconded and passed that we send $20 to Marty for the program. John will communicate with Marty about several problems he has encountered.

NEW BUSINESS: 1. John Johnson announced he is working on an A/L sub program to enlarge sprites. Look for more information in the newsletter.

DISCUSSION: 1. John Johnson would like to buy VIDEO CHESS. 2. According to Jim Reiss the Ottawa TI-FEST went well but PRESS did not make it there.

DEMONSTRATION: Jerry Canady shared his interest in THE EXPLORER VN1.0. This is a memory editor and good learning tool. It is excellent to show how a program works and how the computer works.

Submitted by Bill Paeth, Secretary

VICE SQUAWKS - - - -

by Gary Bishop

I have noticed at our meetings that things sometimes don't proceed as smooth as we would like. Now, I'm not saying I know all the answers on how to fix it, but for the moment, I will just pipedream, and see where it takes us.

In my opinion, an ideal meeting would consist of two meeting rooms, one with a presentation, and one with the club's system and library. This would allow persons that want to copy programs from the library to do so without disturbing the presenter. People could quietly slip in and between meetings. This way, all the bull sessions, questions, assistance, etc. could take place without the constraint of time. I feel we need at least one free hour to accomodate our members, in regards to copying files from the library. Also, I would like to have maybe one half hour more for the meeting, because I never seem to be able to get everything done that needs done, nor copying what I want from the library. I have noticed from the newsletters a popular meeting time is

Saturday morning, or Sunday afternoon. I am not suggesting we change our meeting day, but it could be considered if it could improve the meeting.

A place with just a little more room would be nice. I'm not knocking our present hosts, because we are enjoying a great deal from them, and I hope they are not put off by us. After all, the price is right: free!

The last off-site meeting in Dubuque was very enjoyable. It was gratifying to see what the lowly ole TI has wrought. It makes me wonder if we could even have two meetings a month: one just for gab, library copying, hardware sessions, and whatever. The other could be the formal business meeting, with the usual program presentation. The informal meeting need not be held at the same location as now, although I'm sure our present hosts would gladly ablige. What do you think? Do you have any other constructive comments that could improve the meetings? If so, please don't keep them to yourself. In general, we are not a bashfull bunch, so let's hear some comments. If you think I'm all wet, tell me so. At least it would be proof that someone reads the articles in the newsletter!

```
70 CALL CLEAR
80 PRINT "FILE CONVERSION PROGRAM","CONVERTS A D/V80 FILE TO    A D/V 132 FILE"
90 PRINT "SO IT CAN BE PRINTED IN     COMPRESSED MODE BY OTHER    PROGRAMS." ::
PRINT :: PRINT
100 ! CONVERT FROM D/V 80 FILE TO D/V 132 FILE, FOR COMPRESSED PRINTING
110 ! BY GARY D. BISHOP CEDAR VALLEY 99ER UG, IOWA
115 ! PLACED IN THE PUBLIC DOMAIN 4-30-89
120 DIM A$(100)
130 INPUT "INPUT FILENAME?:                ":I$
140 OPEN #1:I$,DISPLAY ,VARIABLE 80,INPUT
150 INPUT "OUTPUT FILENAME?:               ":O$
160 OPEN #2:O$,DISPLAY ,VARIABLE 132,OUTPUT
170 B=-1
180 LAST=100
190 FOR L=1 TO 100
200 IF EOF(1)THEN LAST=L :: GOTO 230 ELSE 210
210 LINPUT #1:A$(L)
220 NEXT L
230 B=B+1
240 PRINT "CONVERTED ";B*100+LAST;" LINES"
250 FOR J=1 TO LAST
260 PRINT #2:A$(J)
270 NEXT J
280 IF EOF(1)THEN STOP ELSE GOTO 190
    CLOSE #1 : CLOSE #2
```

MAXISPRITE DOCS

The program on the following two pages demonstrates a method of grouping sprites together to form what I call maxisprites. For a shorter version enter only lines 100,190,280-320,850-890,900-940. Enter also line 390 but delete both GOSUB 830 parts add ::END at it's end. The whole program these instructions are in the club library on disk #373. (Maxisprites)

To visualize maxisprites look at the call magnify 4 section in the extended basic book. There each sprite is made up of 4 sections. In a maxisprite each of those sections is a sprite. You can use either X3 or X4 magnification but you must space the sprites accordingly. Times 3 magnification sprites will be 16 dots apart and X4 will be 32. The assembly program to start the sprites all together is in lines 900-940. The call loads in lines 870 803 pass values to the assmembly program. At speeds over 10 maxisprites can show gaps. Those made with 4 sprites can be spaced at 15 or 31 dots apart to overcome this. Larger ones may get more than 5 sprites in a row and blank out. To display maxisprites faster I use call color to reveal them. Note that they must be created with an original speed of 0,0. Also they must be created entirely above dot row 192.

Example program- Line 100 has call init gosub 900 to load the assembly program. The subprogram 850-890 called in line 320 creates maxisprites at X3
magnification with 4 sprites per maxisprite. In line 390 these are changed to X4 magnification. Note that D is changed to 32 to position the sprites correctly. They are called into motion and revealed to the veiwer in line 890. The snowball in the next part starts out as a normal sprite with changes in magnification making it larger. Lines 720-750 create the maxisprite for the final large versions of the snowball. At first the inner 4 sprites of this 16 sprite maxisprite are defined as a smaller snowball while the outer 12 sprites are not revealed. The inner sprites are revealed in line 762. Then in line 770 the whole sprite is revealed the inner 4 sprites chars are defined as solid white. The final scene consists of sending the outer 12 sprites outward in line 802 and redefining the inner sprites to the smaller snowball in line 804.

To use the assembly progam first CALL INIT in your program and then use a GOSUB to load it. (lines 900-940) After you have formed your maxisprites in extended basic load your values into the assembly program per line 870 as follows. CALL LOAD (9984,X speed,Y speed,sprite #,maxisprite quantity). The # 9984 is the address that you are poking your values to. The X Y speeds are the same as dot speeds for regular sprites. The sprite # is the first sprite # you will use. After the first maxisprite other sprites must be either 4 or 16 #s higher or you will overwrite lower # maxisprites. The last item is a 1 for a 4 sprite maxisprite and 2 for a 16 sprite one. Finally enter CALL LINK ("MOTION") when you want your maxisprites to move.

I hope you can use these larger sprites. Now I need a way to make those snowballs grow more gradually.....

J Johnson CR

MAXISPRITES

```
100 CALL CLEAR :: CALL SCREEN(6):: CALL INIT :: GOSUB 900
110 CALL CLEAR :: CALL SCREEN(6)
120 DISPLAY AT(23,1):"WITHOUT THIS PROGRAM SPRITESWERE SMALL."
130 RANDOMIZE :: FOR I=1 TO 14 :: CALL SPRITE(#I,46,16,RND*255+1,RND*191+1,15+(R
ND*(-10)),-5+(RND*5)):: NEXT I
140 CALL COLOR(11,1,16):: CALL CHAR(112,""):: CALL CHAR(101,"085A2CFA5F345A10")
150 F=24 :: GOSUB 830
160 DISPLAY AT(22,1):"THEY COULD BE LARGER.                    " :: FOR I=1
TO 14 :: CALL PATTERN(#I,101):: NEXT I
170 CALL CHAR(128,"010123191B052BFF2B03051B19230101000084189820D4FF9440209818840
000")
180 CALL CHAR(40,"0F1F1F3F3F3F7F7F7F7F7FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FF")
190 FOR I=1 TO 14 :: CALL COLOR(#I,1):: CALL PATTERN(#I,128):: NEXT I :: CALL MA
GNIFY(3):: FOR I=1 TO 14 :: CALL COLOR(#I,16):: NEXT I
200 CALL CHAR(48,"0F0703030100000000000000000000FFFFFFFFFFFF7F3F1F070100000000
00")
210 DISPLAY AT(20,1):"AND LARGER STILL.                        " :: F=23
220 GOSUB 830 :: CALL MAGNIFY(4):: F=22 :: GOSUB 830 :: GOSUB 820
230 CALL CHAR(52,"0001073FFFFFFFFFFFFFFFFFFFFFFF0FFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FF")
240 CALL CHAR(56,"FFFFFFFFFFFFFFFFFFFFFFFF3F0F0100FFFFFFFFFFFFFFFFFFFFFFFFFFFF
0F")
250 DISPLAY AT(20,1):"BUT NO LARGER THAN THIS.                 "
260 GOSUB 820 :: CALL CHAR(60,"F0FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF0080E0FCFFFFFFFF
FFFFFFFFFFFFFFFF")
270 F=21 :: GOSUB 830 :: DISPLAY AT(19,1):"LETS'S GO TO MAXI SPRITES.
                "
280 CALL DELSPRITE(ALL):: CALL MAGNIFY(3)
290 CALL CHAR(128,"0000000008040201060600101039 11FF0808050F65C1C107B571E3DB1B876
BFF")
300 CALL CHAR(132,"7F39101000080606010204080000000FF8F07DBD923731905C7C1250F050
101")
310 CALL CHAR(136,"0080C0F8C28181CCCD828599D8E0F6FF0000000010A0C0E060108088 08DC4
8FF")
320 CALL CHAR(140,"FFD0E0DB9B849A98C0E383C4F0C08000FFCE4810002060608040201000000
000"):: D=16 :: GOSUB 850
380 GOSUB 810 :: DISPLAY AT(17,1):"OR GO TO REALLY BIG FLAKES.
      "
390 F=20 :: GOSUB 830 :: F=19 :: GOSUB 830 :: CALL DELSPRITE(ALL):: CALL MAGNIFY
(4):: D=32 :: GOSUB 850
450 F=20 :: GOSUB 830 :: F=19 :: GOSUB 830 :: FOR F=18 TO 1 STEP -1
470 GOSUB 840 :: GOSUB 830 :: NEXT F :: GOSUB 840 :: CALL DELSPRITE(ALL)
480 CALL CHAR(72,"FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF0F8F8FCFCFCFEFEFEFEFEFFFFFFFF
FF")
490 FOR I=132 TO 135 :: CALL CHAR(I,"FFFFFFFFFFFFFFFF"):: NEXT I :: CALL COLOR(1
1,1,1)
500 CALL CLEAR :: FOR F=24 TO 12 STEP -1 :: GOSUB 830 :: NEXT F :: CALL CHAR(112
,""):: CALL COLOR(11,2,15,10,2,1)
510 CALL CHAR(76,"FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFEFEFEFEFEFCFCF8F8F0
F0")
520 CALL CHAR(80,"FFFFFFFFFFFFFEFCF8E080000000000F0E0C0C0800000000000000000000
00")
530 CALL CHAR(64,"FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF0FFFFFFFFFFFFFFFFFFFFFFFFFEE010
00")
540 CALL CHAR(36,"00000000000000000000001030307 0F000000000001071F3F7FFFFFFFFFF
FF")
```

```
550 CALL CHAR(68,"0000000000080E0F1FCFEFFFFFFFFFFFF0000000000000000000000080C0C0E0
F0")
560 CALL CHAR(104,"0018183C3C3C1838"):: CALL CHAR(105,"98583C3C3D181818"):: CALL
 MAGNIFY(1)
570 CALL CHAR(106,"0018183C3C3C1838")
580 DISPLAY AT(11,26):"h" :: CALL CHAR(84,"0000000103070F1F1F3F3F7F7F7FFFFF031F7
FFFFFFFFFFFFFFFFFFFFFFFFFFF")
590 DISPLAY AT(11,26):"i" :: CALL SPRITE(#1,46,16,75,211,-1,-2)
600 CALL CHAR(88,"FFFF7F7F7F3F3F1F1F0F07030100000FFFFFFFFFFFFFFFFFFFFFFFF7F1F
03")
610 DISPLAY AT(11,26):"j" :: CALL CHAR(92,"C0F8FEFFFFFFFFFFFFFFFFFFFFFFFFFF00000
080C0E0F0F8F8FCFCFEFEFEFFFF")
620 CALL CHAR(46,"187E7EFFFF7E7E18")
630 CALL CHAR(96,"FFFFFFFFFFFFFFFFFFFFFFFFFFFFEF8C0FFFFFEFEFEFCFCF8F8F0E0C080")
640 CALL CHAR(128,"03071F3F7F7FFFFFFFFF7F7F3F1F0F03C0E0F8FCFEFEFFFFFFFFFEFEFCF8F
0C0"):: CALL PATTERN(#1,128)
650 CALL MAGNIFY(3):: CALL MOTION(#1,-1,-3):: GOSUB 810 :: CALL MAGNIFY(4):: CAL
L MOTION(#1,-1,-4)
660 CALL CHAR(44,"FFFFFFFFFF7F7F7F7F7F3F3F3F1F1F0FFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FF")
670 CALL POSITION(#1,X,Y):: Y=Y-30 :: X=X-10
720 CALL SPRITE(#2,84,1,X,Y,0,0,#3,88,1,X+32,Y,0,0,#4,92,1,X,Y+32,0,0,#5,96,1,X+
32,Y+32,0,0)
730 CALL SPRITE(#6,36,1,X-32,Y-32,0,0,#7,40,1,X,Y-32,0,0,#8,44,1,X+32,Y-32,0,0,#
9,48,1,X+64,Y-32,0,0)
740 CALL SPRITE(#10,52,1,X-32,Y,0,0,#11,56,1,X+64,Y,0,0,#12,60,1,X-32,Y+32,0,0,#
13,64,1,X+64,Y+32,0,0)
750 CALL SPRITE(#14,68,1,X-32,Y+64,0,0,#15,72,1,X,Y+64,0,0,#16,76,1,X+32,Y+64,0,
0,#17,80,1,X+64,Y+64,0,0)
760 CALL LOAD(9984,-2,-6,2,2):: CALL LINK("MOTION")
762 CALL COLOR(#2,16,#3,16,#4,16,#5,16):: CALL DELSPRITE(#1):: GOSUB 810
770 CALL COLOR(#6,16,#7,16,#8,16,#9,16,#10,16,#11,16,#12,16,#13,16,#14,16,#15,16
,#16,16,#17,16)
800 CALL PATTERN(#2,132,#3,132,#4,132,#5,132):: GOSUB 810 :: CALL LOAD(9984,0,0)
:: CALL LINK("MOTION")
802 CALL MOTION(#6,-50,-55,#9,25,-48,#17,40,60,#14,-70,57,#10,-60,0,#12,-65,0,#1
1,80,0,#13,70,0)
803 CALL MOTION(#7,0,-60,#8,0,-54,#15,0,63,#16,0,40)
804 CALL PATTERN(#2,84,#3,88,#4,92,#5,96)
806 CALL DELSPRITE(#10,#12,#11,#13,#7,#8,#15,#16,#6,#9,#14,#17):: GOSUB 820 :: C
ALL COLOR(#2,1,#3,1,#4,1,#5,1)
808 CALL DELSPRITE(ALL):: GOSUB 820 :: CALL CLEAR :: CALL MAGNIFY(1):: CALL CHAR
SET :: GOTO 110
810 FOR I=1 TO 200 :: NEXT I :: RETURN
820 FOR I=1 TO 500 :: NEXT I :: RETURN
830 CALL HCHAR(F,1,112,2):: DISPLAY AT(F,1):"pppppppppppppppppppppppppppp" :: CA
LL HCHAR(F,28,112,5):: RETURN
840 FOR I=1 TO 40 :: NEXT I :: RETURN
850 FOR I=0 TO 5 :: A=RND*160+1 :: B=1+I*4
860 CALL SPRITE(#B,128,1,A,A,0,0,#B+1,132,1,A+D,A,0,0,#B+2,136,1,A,A+D,0,0,#B+3,
140,1,A+D,A+D,0,0)
870 CALL LOAD(9984,15+(RND*(-10)),-5+(RND*5),B,1)
890 CALL LINK("MOTION"):: CALL COLOR(#B,16,#B+1,16,#B+2,16,#B+3,16):: NEXT I ::
RETURN
900 CALL LOAD(10020,2,224,39,4,193,32,39,2,9,132,6,4,10,36,193,96,39,2,10,133,9,
101,2,133,0,4)
910 CALL LOAD(10046,19,2,2,5,0,16,4,195,2,0,7,128,160,4,2,1,39,0,2,2,0,2,4,32,32
,36,2,32,0,4)
920 CALL LOAD(10076,5,131,129,67,17,249,4,224,131,124,2,224,131,224,4,96,0,112)
930 CALL PEEK(8196,AA,CC):: BB=AA*256+CC :: BB=BB-8 :: DD=INT(BB/256):: EE=(BB-I
NT(BB))*256
940 CALL LOAD(8196,DD,EE):: CALL LOAD(BB,77,79,84,73,79,78,39,36):: RETURN
```

TIPS FROM THE TIGERCUB

#52

TIGERCUB SOFTWARE
156 Collingwood Ave.
Columbus, OH 43213

Over 120 original programs in Basic and Extended Basic, available on cassette or disk, NOW REDUCED TO JUST $1.00 EACH!, plus $1.50 per order for cassette or disk and PP&H. Minimum order of $10.00. Cassette programs will not be available after my present stock of blanks is exhausted. The Handy Dandy series, and Color Programming Tutor, are no longer available on cassette.
Descriptive catalogs, while they last, $1.00 which is deductable from your first order.

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

NUTS & BOLTS DISKS
These are full disks of 100 or more utility subprograms in MERGE format, which you can merge into your own programs and use, almost like having another hundred CALLs available in Extended Basic. Each is accompanied by printed documentation giving an example of the use of each. NUTS & BOLTS (No. 1) has 100 subprograms, a tutorial on using them, and 5 pp. documentation. NUTS & BOLTS No. 2 has 108 subprograms, 10 pp of documentation. NUTS & BOLTS #3 has 140 subprograms and 11 pp. of documentation. NOW JUST $15 EACH, POSTPAID.

TIPS FROM THE TIGERCUB
These are full disks which contain the programs and routines from the Tips from the Tigercub newsletters, in ready-to-run program format, plus text files of tips and instructions.
TIPS (Vol. 1) contains 50 original programs and files from Tips newsletters No. 1 through No. 14. TIPS VOL. 2 contains over 60 programs and files from Nos. 15 thru 24. TIPS VOL. 3 has another 62 from Nos. 25 through 32. TIPS VOL. 4 has 48 more from issues No. 33 through 41.
NOW JUST $10 EACH, POSTPAID.

```
*****************************
*        NOW READY          *
* TIPS FROM TIGERCUB VOL.5  *
* Another 49 programs and   *
* files from issues No. 42  *
* through 50. Also $10 ppd  *
*****************************
```

TIGERCUB CARE DISKS #1,#2,#3 and #4. Full disks of text files (printer required).
No. 1 contains the Tips newsletters #42 thru #45, etc. Nos. 2 and 3 have articles mostly on Extended Basic programming. No. 4 contains Tips newsletters Nos. 46-52. These were prepared for user group newsletter editors but are available to anyone else for $5 each postpaid.

This one should come in handy for bowling league captains and Little League coaches.

```
100 DIM M(29,29),T$(30)
110 GOTO 130
120 N;Q$;J;I;X;P$;S$;K
130 !@P-
140 DISPLAY AT(3,7)ERASE ALL
:"LEAGUE SCHEDULER":;:"by th
e Burwells          adapt
ed by Tigercub"
150 DISPLAY AT(8,1):" This p
rogram sets up a":"schedule
 for up to 30 teams":"so that
 each plays each":"other onc
e and only once."
160 DISPLAY AT(12,1):" If an
 odd number of teams":"are s
cheduled, each gets one":"by
e."
170 DISPLAY AT(16,1):"Number
 of teams?" :: ACCEPT AT(16,
18)VALIDATE(DIGIT):N :: IF N
>30 THEN DISPLAY AT(18,1):"L
IMIT OF 30!" :: GOTO 170
180 DISPLAY AT(18,1)ERASE AL
L:"Schedule teams by name? Y
" :: ACCEPT AT(18,25)SIZE(-1
)VALIDATE("YN"):Q$ :: IF Q$=
"N" THEN 200
190 FOR J=1 TO N :: DISPLAY
AT(20,1):"Team no.";J;"name?
" :: ACCEPT AT(22,1):T$(J)::
 NEXT J :: GOTO 210
200 FOR J=1 TO N :: T$(J)="T
eam No. "&STR$(J):: NEXT J
210 IF N/2<>INT(N/2)THEN N=N
+1 :: T$(N)="bye"
220 DISPLAY AT(23,1):"Schedu
le by day, week, month":"or
what?" :: ACCEPT AT(24,10):S
$ :: FOR J=1 TO N-1 :: M(1,J
)=J+1
230 NEXT J :: FOR J=1 TO N-1
 STEP 2 :: GOSUB 260
240 NEXT J :: FOR J=2 TO N-2
 STEP 2 :: GOSUB 330
250 NEXT J :: GOSUB 390 :: S
TOP
260 FOR I=1 TO N-2 :: IF M(I
,J)=N THEN 280
270 M(I+1,J)=M(I,J)+1 :: GOT
O 290
280 M(I+1,J)=M(I,J):: GOTO 3
00
290 NEXT I
300 X=I+1 :: FOR I=X TO N-2
:: M(I+1,J)=M(I,J)-1
310 NEXT I
320 RETURN
330 FOR I=1 TO N-2 :: IF M(I
,J)=2 THEN 350
340 M(I+1,J)=M(I,J)-1 :: GOT
O 360
350 M(I+1,J)=M(I,J):: GOTO 3
70
360 NEXT I
370 X=I+1 :: FOR I=X TO N-2
:: M(I+1,J)=M(I,J)+1
380 NEXT I :: RETURN
390 DISPLAY AT(12,1)ERASE AL
L:"Output to - 2":;:" (1) Sc
reen":" (2) Printer" :: ACCE
PT AT(12,13)SIZE(-1)VALIDATE
("12"):K :: IF K=1 THEN 440
400 DISPLAY AT(18,1):"Printe
r? PIO" :: ACCEPT AT(18,10)S
IZE(-18):P$ :: OPEN #1:P$ ::
 PRINT #1:"LEAGUE SCHEDULE":
: :: FOR I=1 TO N-1 :: PRIN
T #1:S$;" #";I :: PRINT #1:T
$(1);" vs ";T$(M(I,1))
410 FOR J=2 TO N-2 STEP 2 ::
 PRINT #1:T$(M(I,J));" vs ";
T$(M(I,J+1))
420 NEXT J :: PRINT #1:"": :
430 NEXT I :: RETURN
440 FOR I=1 TO N-1 :: PRINT
TAB(7);"LEAGUE SCHEDULE": :
:: PRINT "WEEK #";I: : :: PR
INT T$(1);" vs ";T$(M(I,1)):
: FOR J=2 TO N-2 STEP 2 :: P
RINT T$(M(I,J));" vs ";T$(M(
I,J+1))
450 NEXT J :: PRINT "": : ::
 PRINT "PRESS ANY KEY FOR NE
XT WEEK"
460 CALL KEY(0,K,S):: IF S=0
 THEN 460
470 CALL CLEAR
480 NEXT I :: RETURN :: END
```

Some folks seem to think that the subprograms on my Nuts & Bolts disks are just flashy screen displays. Not so! This one will be on the next diskfull, if I ever get it full, which is most unlikely.
ACCEPT AT with a negative

size is useful to accept a default string from the screen, but the length of the string is limited to 28 characters; and if you want something other than the default, you must be sure to delete any extra characters.
CALL DEFAULT(R,C,M$,R$),
where R and C are the row and column to accept at, M$ is the default string which can be up to 254 characters long, and R$ is the string accepted, will display the default string, accept it if Enter is pressed, or accept any other string without having to blank out the extra characters. Just don't type too fast!

```
100 M$="TESTING" :: CALL CLE
AR
110 CALL DEFAULT(12,1,M$,R$)
:: DISPLAY AT(24,1):R$ :: GO
TO 110
10000 SUB DEFAULT(R,C,M$,R$)
:: R$="" :: X=ASC(M$)
10001 DISPLAY AT(R,C):M$
10002 CALL HCHAR(R,C+2,ASC(S
EG$(M$,1,1))):: CALL HCHAR(R
,C+2,30)
10003 CALL KEY(0,K,S):: IF S
=0 THEN 10002 ELSE IF K=13 T
HEN R$=M$ :: SUBEXIT ELSE DI
SPLAY AT(R,C):CHR$(K):: ACCE
PT AT(R,C+1):R$ :: R$=CHR$(K
)&R$
10004 SUBEND
```

CALL DEFAULT(R,C,N,RN), with N as the default value and RN as the value accepted, will do the same for numeric input, and will reject any non-numeric input. Errors due to fast typing can be prevented by omitting the DISPLAY AT(R,C):CHR$(K) in line 1002.

```
100 N=176453.897 :: CALL CLE
AR
110 CALL DEFAULTN(12,1,N,RN)
:: DISPLAY AT(24,1):RN :: GO
TO 9999
10000 SUB DEFAULTN(R,C,N,RN)
:: DISPLAY AT(R,C):N :: N$=S
EG$(STR$(N),1,1)
10001 CALL HCHAR(R,C+2,ASC(N
$)):: CALL HCHAR(R,C+2,30)
10002 CALL KEY(0,K,S):: IF S
=0 THEN 10001 ELSE IF K=13 T
HEN RN=N :: SUBEXIT ELSE DIS
PLAY AT(R,C):CHR$(K):: ACCEP
T AT(R,C+1):R$ :: R$=CHR$(K)
&R$
10003 ON ERROR 10004 :: RN=V
AL(R$):: GOTO 10005
10004 CALL SOUND(200,110,5,-
4,5):: DISPLAY AT(R,C):N ::
ON ERROR STOP :: RETURN 1000
2
10005 SUBEND
```

Ed Machonis discovered an easy way to count the words in a TI-Writer file, using TI-Writer itself. Just put in a line before line 0001, with .LM 0;RM 1;FI;PL nnn with nnn being the sector length of the file multiplied by 40. Save it, go into the Formatter and print it to disk under a different filename. Return to Editor, load the resulting file, page through it with FCTN 4 counting any blank lines, subtract the number of blanks from the last line number, and that's it! The Formatter takes about one minute to count 1000 words. If the resulting file is very large, you may have to load it in two sections.

```
100 M$="POS WILL FIND THE FI
RST OCCURRENCE OF A SUBSTRIN
G WITHIN A STRING BUT I OFTE
N NEED TO FIND THE LAST OCCU
RRENCE SO I WROTE THIS SUBPR
OGRAM"
105 INPUT "SUBSTRING?":L$
110 CALL LAST(M$,L$,P):: IF
P=0 THEN PRINT "NOT FOUND" :
: GOTO 105 ELSE PRINT SEG$(M
$,P,255):: GOTO 105
120 SUB LAST(M$,L$,P):: X=1
130 Y=POS(M$,L$,X):: IF Y=0
THEN P=0 :: SUBEXIT ELSE Z=Y
140 X=Y+1 :: Y=POS(M$,L$,X):
: IF Y=0 THEN P=Z :: SUBEXIT
 ELSE Z=Y :: GOTO 140
150 SUBEND
```

Here's a new way to make music. The algorithm in 110 sets up a 3-octave chromatic scale - note the N(1)=F, I have erroneously omitted it when I previously published that algorithm.
To change the key of the music you have programmed, just change the value of F. Lines 190-220 contain the part of the music that is repeated within the melody.
A is the subscript of the melody note, B is the subscript number of the chord. These must be above 13, as the frequency is divided by 2 in the subroutine.
Each beat of the music has a GOSUB, to 230 to play a bass accompaniment with the first note of each bar, to 260 for the other notes of the bar. The chord note is divided by different values to play the three notes of the chord in succession, and multiplied by 3.75 in the 3rd voice to produce a bass note two octaves lower in the -4 noise. The melody note is multiplied by 1.01 in the second voice to give a richer tone.

```
100 DISPLAY AT(12,3)ERASE AL
L:"THE MAORI FAREWELL SONG"
! programmed by
  Jim Peterson
110 F=110 :: DIM N(36):: FOR
 J=1 TO 36 :: N(J)=INT(F*1.0
59463094^(J-1)):: NEXT J ::
N(1)=F :: T=-999
120 GOSUB 190 :: A=30 :: B=2
3 :: GOSUB 230 :: GOSUB 260
:: GOSUB 260 :: A=32 :: B=28
 :: GOSUB 230 :: GOSUB 260 :
: GOSUB 260 :: A=28
130 GOSUB 230 :: GOSUB 260 :
: GOSUB 260 :: A=30 :: B=23
:: GOSUB 230 :: GOSUB 260 ::
 A=28 :: GOSUB 260 :: A=27 :
: GOSUB 230 :: GOSUB 260
140 A=28 :: GOSUB 260 :: A=3
0 :: GOSUB 230 :: GOSUB 260
:: GOSUB 260 :: GOSUB 230 ::
 GOSUB 260 :: GOSUB 260 :: G
OSUB 190
150 A=30 :: B=23 :: GOSUB 23
0 :: GOSUB 260 :: GOSUB 260
:: A=32 :: B=16 :: GOSUB 230
 :: GOSUB 260 :: A=28 :: GOS
UB 260
160 A=33 :: B=23 :: GOSUB 23
0 :: GOSUB 260 :: A=32 :: GO
SUB 260 :: A=25 :: B=13 :: G
OSUB 230 :: GOSUB 260 :: GOS
UB 260
170 A=27 :: B=23 :: GOSUB 23
0 :: GOSUB 260 :: GOSUB 260
:: A=28 :: B=16 :: GOSUB 230
 :: GOSUB 260 :: GOSUB 260
180 B=28 :: GOSUB 230 :: GOS
UB 260 :: GOSUB 260 :: B=16
:: GOSUB 230 :: GOSUB 260 ::
 GOSUB 260 :: GOTO 120
190 A=32 :: B=28 :: GOSUB 23
0 :: GOSUB 260 :: GOSUB 260
:: A=28 :: B=16 :: GOSUB 230
 :: GOSUB 260 :: A=30 :: GOS
UB 260
200 A=32 :: B=28 :: GOSUB 23
0 :: GOSUB 260 :: GOSUB 260
:: B=16 :: GOSUB 230 :: GOSU
B 260 :: GOSUB 260 :: B=28 :
: GOSUB 230 :: GOSUB 260
210 A=30 :: GOSUB 260 :: A=3
3 :: B=23 :: GOSUB 230 :: GO
SUB 260 :: A=27 :: GOSUB 260
 :: A=28 :: B=16 :: GOSUB 23
0 :: GOSUB 260 :: GOSUB 260
220 B=28 :: GOSUB 230 :: GOS
UB 260 :: GOSUB 260 :: B=16
:: GOSUB 230 :: GOSUB 260 ::
 GOSUB 260 :: RETURN
230 CALL SOUND(T,N(A),5,N(B)
/1.585,9,N(B)*3.75,30,-4,9):
: GOSUB 290
240 CALL SOUND(T,N(A),5,N(B)
/1.334,9,N(B)*3.75,30,-4,9):
: GOSUB 290
250 CALL SOUND(T,N(A),5,N(B)
/2,9,N(B)*3.75,30,-4,9):: GO
SUB 290 :: RETURN
260 CALL SOUND(T,N(A),5,N(A)
*1.01,5,N(B)/1.585,9):: GOSU
B 290
270 CALL SOUND(T,N(A),5,N(A)
*1.01,5,N(B)/1.334,9):: GOSU
B 290
280 CALL SOUND(T,N(A),5,N(A)
*1.01,5,N(B)/2,9)
290 FOR D=1 TO 20 :: NEXT D
:: RETURN
```

MEMORY FULL.....

Jim Peterson

SECTOR SHARING

by Mark Schafer
BLUEGRASS 99 COMPUTER SOCIETY, INC.

They say necessity is the mother of invention. And in this case, I'm the father. I think I've discovered something you'll find intriguing. Take a look at the following disk catalog:

```
DSK1 - DISKNAME= FNWEB/4#1
 AVAILABLE=   7  USED= 351
FILENAME  SIZE    TYPE     P
---------- ---- ---------- -
  AS         33 PROGRAM
  AT         22 PROGRAM
  CF         31 PROGRAM
  CG         25 PROGRAM
  CHARA1      5 PROGRAM
  D1         33 PROGRAM
  D2         33 PROGRAM
  D3         29 PROGRAM
  DU         33 PROGRAM
  DV         33 PROGRAM
  DW         29 PROGRAM
  EA          9 PROGRAM
  ED         33 PROGRAM
  EE         19 PROGRAM
  LH         12 PROGRAM
  LOAD       31 PROGRAM
  QD         12 PROGRAM
  SL         10 PROGRAM
  SYSCON      6 PROGRAM
  UL          4 PROGRAM
```

If you don't notice anything strange, add up the sizes of the files and compare that to the number of sectors used. How did I do that? Why did I do that? That's what I'm here to tell you.

The above is the catalog of my Funnelweb disk. What I've done is to make it so that some files take up the same space as other files which is the concept I call sector-sharing.

First, let's get into why I did it. I have Disk Utilities by John Birdwell. One of its features is the ability to change to default system setup. The trouble was sometimes I will want the defaults to be one way, and sometimes I will want them another. Now, I could change the setup in the program when I need to, but this is some trouble. The ideal solution would be to have two (or more) copies of the program on the disk and boot the one with the defaults I want at the time. But I only have one SSSD disk drive, so I clearly don't have the room for this. Just like limited memory can lead to tight coding, limited disk space can lead to creative disk utilization.

All I wanted to do was to change the first sector. So I got the idea to create a file that would have a different first sector, but share the rest of the sectors with the original file!

The steps to do this, I believe, can be done in any order. Basically, it goes like this: creating the new header sectors, creating the modified sector, updating the disk catalog, marking the used sectors, and renaming the new files. The beauty is that Disk Utilities itself can handle all of the above in one session, but I suppose any sector editor and disk manager will do.

So let's create the new header sectors first. Each file on a disk has to have a sector that identifies the type of file it is and where it is on the disk. The first step here is to find out what sectors are free. One way to do this is to look at sector 0 starting at byte>38, look for non-F's, and figure out what sectors correspond to the blank bits. Or you could use Disk Utilities to print the disk report and figure out what sectors are contained in no file. In my case, sectors>13 to>16 were available. You could put them anywhere, but the normal thing is to put header sectors in the>02 to>21 range.

Next, I need to know where on the disk the files I'm going to "copy" are, as well as where their header sectors are. The disk report has this information. So now you edit the header sector of the these files. Go to the first one first. You need to change two things on it. Change the name to something that would fall at the end of the disk catalog. This way, we don't have to insert when we change sector 1. I called my new files ZX, ZY and ZZ. For the file that has the modified sector, you need to change the segments starting at byte>1C. Insert three bytes at this point. This may be a little more difficult with some sector editors. Put in the following three bytes at 1C: yz 0x 00, where xyz is the sector we're going to create in step 2. In my case, it was>16, so I inserted 16 00 00. Then add one to the next byte, so if it's>57, make it>58. This process makes it so this file is in the same place as the original file except its first sector is different. If you're changing a sector in the middle, this is a bit more difficult. When you save it back, put it at the first available sector you found. For the remaining header sectors, I just simply changed their names and saved them to next available sectors since they are to share exactly the same sectors.

Once you've got that done, the rest of it is a cinch. To create the modified sector, simply edit the sector you wish to change, make the appropriate changes, and save it to the free sector you indicated at step 1 (#16 in my case.) Normally the sectors contained in a file are higher than>21, but I didn't have any free in that area.

Next, it's time to change the disk catalog at sector 1. Simply put the header sectors you created in step 1 at the first available 0000 in sector 1. I appended 0013 0014 0015 to add my three new files.

The next step is to tell the disk what sectors we've used. With disk Utilities, you just use the Mark Sector feature. With others, you may have to figure out what bits they correspond to in sector 0 and make the changes yourself.

The last step is to rename the new files what you really want to call them. I called mine D1, D2, and D3. The last bytes have to be consecutive so that they load as one continuous program. After this, I had to configure Funnelweb to be able to load my new programs. So now, when I run Disk Utilities, I have the choice of the options in DU or the options in D1. They both load just fine.

But there are some consequences. There's the problem of copying. If you try to copy a sector-sharing disk by file, the duplicate will unshare them. Also, you may get an out-of-space error. So to copy such a disk, you should use a sector copier. Then there's the problem of what happens if you want to copy the sector-sharing files, but not the whole disk. If you can't do a direct copy sector x to sector x, I would recommend that you find a way to sector copy the whole disk and delete the files you didn't want. Or you could copy only one of them and start the operation over again on the new disk.

So to make a copy of this 95-sector program, it took only 4 additional sectors: 3 for each new header sector, and 1 for the modified sector. Shorter files would need even fewer additional sectors. I could go on and make another version of this program, but I think I'm happy with just two. I wonder if I have any other files I can do this to...

NEXT MEETING

MONDAY, JUNE 12

6:30 PM --- WEST MUSIC COMPANY

FINISH THE LAWN EARLY, SO YOU

DON'T MISS ANY OF THE ACTION!

COME AS YOU ARE: FORMAL ATTIRE

NOT REQUIRED FOR ENTRY.

Cedar Valley 99'er Users Group
288 Windsor Dr. NE
Cedar Rapids, Iowa 52402

Send To:

GARY BISHOP
124-222
3270 28TH AVE
MARION IA 52302