



Ogden's 99'er User Group  
 1396 LINCOLN AVE. APT. B OGDEN UTAH 84404  
 GROUP OFFICERS

President: Jim Buck 773-2552  
 Vice President: Harold Bingham 394-6362  
 Secretary-Treasurer: Richard Scott 776-2551  
 Librarian: Harold Kilburn 773-0622  
 Newsletter Editor: Neil Brown 393-9605  
 Associate Editor: Harold Kilburn 773-0622

**APRIL 1990 NEWSLETTER  
 OUR NEXT MEETINGS ARE:**

SATURDAY, APRIL 07 TIME: 0900 hrs.  
 TUESDAY, APRIL 03 TIME: 1900 hrs.  
 TUESDAY, APRIL 17 TIME: 1900 hrs.

We will be meeting in the CIVIL AIR PATROL building at the OGDEN MUNICIPAL AIRPORT RAMPART ROAD. On the 03rd and the 17th we have started a BASIC class for those interested in the people taking the BASIC class will meet on the FIRST and THIRD Tuesday of the month at times above. Jim Buck is the teacher. These meetings are held at the Hill Side Center in the basement. Roy City Center.

QUEEN TI USERS GROUP  
 1396 LINCOLN APT B  
 OGDEN, UTAH 84404

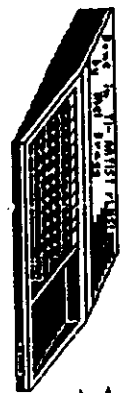
HAPPY EASTER  
 ENVOYED BY  
 1989 W. JAMES  
 LAYTON, UT 84403

Happy Easter



A Publication of the TI 99/4 and 9640  
 Ogden Users Group Inc.

**TI 99-4  
 ORDER  
 USERS  
 GROUP**



THE CLUB THAT REFUSES  
 TO SAY GOODBYE



Holly Bahasapirats  
 I must have taken the wrong  
 turn back there!

APRIL 1990

15  
 J86

*Handwritten note:* 13 May 1990

TI BASIC FOR BEGINNERS  
 by Harold Hiburn  
 CHAPTER TWO OF SIMPLE BASIC  
 PROGRAMMING

In the previous sessions, we were in the immediate BASIC statements with the CALL CLEAR, LET, and CALL SOUND. CALL VCHAR, and CALL PRINT. But now it is time to do something. Lost you after immediately. Knowledge of the way the computer must work with the math problems. Now we will start putting some of this together and produce "programs".

Remember, we all do our best learning from making mistakes, so get ready for your "real learning".

You remember the exercise, PRINT "HI THERE?". The computer did it immediately. In programming, we can store statements in memory for later use. A program line must start with a space number followed by a space as follows. But first type in the Command NEW and PRINT. This will clear the computer's memory of any program already in memory, and clears the screen. It is different than CALL CLEAR which only clears the screen.

10 PRINT "HI THERE?"  
 20 PRINT "ARE YOU READY?"  
 30 PRINT "TO LEARN BASIC?"  
 40 END

If errors are made, there are several ways to correct them. If you are still on the line, just use the arrow keys to correct the error. If you have already entered the line, then just retype it completely and the new line and number will replace the bad one.

Or you can type in the number and then press the key for X (the key for the computer will then display the bad line with the cursor over the text of the line for you to make the corrections. After correcting, press ENTER and the corrected line is now a part of your program.

Do you want to see the program again? Then just type in LIST. After this simple four line program is typed in correctly, it can be operated by typing in the Command RUN followed by ENTER. Try it and try it again.

Now let's modify this program a bit. In running it several times we have cluttered the screen with it and the RUN command. It is so simple to improve programs with BASIC that you will do all kinds of things to future programs and really like to do it. Particularly when it works well, you want all the clutter each time you add the following:

```

>: LINE CLINK
PRESS ENTER and this is new part of your program to check it, just call for LIST. S is the line number (call program elements must be in a numbered line) and then the CALL CLEAR is an old friend from earlier classes. Now try RUN. Run again and again. Notice while the screen turns green actually running and when returns to the light blue.

```

Lets review the requirements for programming!

Each program line must have a line number followed by a space. Save room between numbers so that new lines can be added between.

Each line must be followed by ENTER to place it in the computer's program storage area.

Each program line will be separated in numerical sequence, regardless of when you type it.

Commands and new statements. We've been using these words as not. Commands, such as NEW, LIST, RUN are immediate and do not need line numbers. You must have the line indicator and the cursor flashing to use immediate commands. If they are not, then clear the program with FCTN 4. The instruction you put in a numbered program line.

If you get lost in where you are, just clear the program (FCTN 4) and LIST, then you'll know where you were. If you don't need that, you can type a NEW and the program memory area will be cleared just as soon as you press ENTER.

Now let's say you want to change a program line. You now you can retype the line including its number. This makes that change. But if BASIC lets you do this much easier. Just type the line number in the immediate area and then press either the up or down arrow (FCTN 8 or FCTN X). Let's try it to change the statement TO ERK BASIC to TO LERN TI BASIC. First, let's use

LIST to get the program visible. Then type the line number you want to change followed by the arrow key. No mistakes. Then use the arrow keys to position the cursor, and in this case, press FKSRT (FCTN 2), make your addition and then press ENTER. Do you want to see it? Use the command LIST or RUN. Try them both to get used to it.

- next month
- CARRIDGES FOR MANAGEMENT \$5.00
- HOUSEHOLD BUDGET ER. CHITSLGHT ER.
- TRAIL \$3.00 CAR WARS 1 ER.
- \$2.00 ER SUPER PERSONAL
- ATRACK \$5.00 ER. PERSONAL
- RECORD KEEPING 2 ER. \$5.00
- ER HUNCHMAN 1 ER. \$2.00 ER
- ER TERMINAL EMULATOR II 1 ER.
- \$5.00 ER ADVENTURE 2 ER.
- \$3.00 ER MICROBURDEN 2
- ER. \$4.00 HOME FINANCIAL
- DECISIONS \$2.00 TOUCH
- TYPING TUTOR \$2.00 DUNKEY
- KONG 1 ER. \$2.00 ER
- PERSONAL RNL. ESTATE \$2.00 ER
- PARSEC ER. \$2.00 ER
- FOOTBALL \$3.00 TI EXTENDED
- BASIC MODULE \$25.00 ER
- TI-WRITER WORD PROCESSOR
- MANUAL \$5.00 TRIPLE TECH
- CARD 64K PRINT BUFFER
- \$65.00 TI PERIPHERAL
- EXPANSION BOX TI DISK
- CONTROLER TI R2232 TI 32K
- MEMORY 2 HALF HEIGHT DRIVES
- DS/DD \$200.00 TI COMPUTER
- KEYBOARDS 6 ER. \$10.00 AT
- CONTRACT NEI. BRAGG AT
- 001(393-9605) DR WRITE 1396
- LINDOLN RUE APT#8 OGDEN. WA-10
- GENEVE \$325.00 TI
- PERIPHERAL EXPANSION BOX TI
- DISK CONTROLER SS/SD DISK
- DRIVE \$ 75.00 CONTRACT
- RICKARD MORRISON
- 001(621-3697) DR WRITE 2570
- FOWLER OGDEN, UTRH 04401

USE THIS TO WRITE LETTERS

If you write a lot of letters, or even just a few and you have [...] writer of one of the clones, the following file may help you. Simply type in the file as you see it listed here and then save it to disk. Any time you want to write a letter just load in the file and you'll be able to write nice neat and properly spaced letters every time.

You will need to customize this file with your address and make any other changes to conform to your style, but this is the suggested format. On the style shown here, we have also included footer and header commands which you might want to include if you are writing long letters or instructions that require more than one page. If you don't want them for your average letter, simply delete lines 0001 and 0003B that include the .NE and .FO.

Once you have customized your file and saved it, simply load it into the editor whenever you want to write a letter. On lines 0013, 0014, 0015, and 0016 you will put in the name and address of who you are writing to. Your greeting on line 0018 and your salutation on line 0026 can be in standard form or you can change it as you like. You can then write your letter adding lines of alternating lines depending on the length. If your entire letter requires more than 58 lines it will automatically use another page.

Don't forget to save your letter after writing it. A handy file name would be the last name of the person your writing.

The format of the headings etc. are taken from the accepted standards used today. With a little experience "LETTERFILE" will find many uses.

```
0001 .NE.....Your Name X
0002 .LN Y3:RM 72:FI:PL 58
0003 .SP 2
0004 Your Name Here
0005 Your Address
0006 Your City and State
0007 Your ZIP code
0008 .SP 2
0009 Your telephone number (optional)
0010 .SP 2
0011 Date xxxxx, 1980
0012 .SP 2:LN 8
0013 To Name
0014 To Address
0015 To City and State
0016 To ZIP code
0017 .SP 2
0018 Greetings,
0019 .SP 2:LN 45
0020 Start your letter here-----
0021
0022
0023
0024 End your letter here-----
0025 .SP 2:LN 43
0026 Your salutation
0027 .SP 5
0028 Your Name
0029 .FO.....Your Name X
0030
```

\*\*\*\*\*
M-AGE/99 & NEW-AGE/
99 & NEW-AGE/99 & N
EM-AGE/99 & NEW-AGE
/99 & NEW-AGE/99 &
\*\*\*\*\*

\* BY JACK SUGRUE, Box 459, East Douglas, MA 01516 \*

There's a new (I think) company making software for the TI: Arcade Action, Program Innovators, 4122 Glenway, Wauwatosa, WI 53222. They have a nice version of TETRIS. They also have the usual space shoot 'ems. They have a program called Snowmobile which is yet another version of the Regena-style skiing/river-travel maneuvering. And they have a bunch of other stuff, all of which is detailed in their free catalog.

But what they have that is WONDERFUL and a true first for the TI Community is a superb cribbage game: It's perfect in the sense that it is completely playable. It is reasonably fast. It is intelligent and colorful and extremely well designed and in Extended Basic for people who like to add their own little touches, though you won't need to add little touches to this game at all. The unknown author even offers simple suggestions for reducing it of its 60 sectors.

There have never been any good, playable TI cribbage games, but this game, called CUTTHROAT CRIBBAGE, is something else entirely. This game not only lets you play against an intelligent computer but allows some interesting inputs. During the game, for example, if you wish to question the computer's judgment (or would like to test out some odd card combos) you simply type "C" for Count Check at the end of a hand (instead of the default "sp" for continuing play). You may then set up a hand, including play card. The computer will score it, explaining each detail. For example, you must type each card with the pip value first and the suit (S,H,D,C) second. A five of Hearts would be 5H; a Queen of Clubs, QCl; an Ace of Diamonds, AD. You get the picture. So take a hand like 5H, JD, 5S, 5C, with the play card the 5D. Any cribbage player's recognize the Grand Hand when they see it and know it equals 29, the highest score in cribbage. The computer will tell you this when you type in that hand. It will also break down all of the patterns by individual scores (15s, Quadruplets, His Nobs) to show you exactly how the hand is scored. You can create any legitimate hand you'd like for detailed analysis by the computer. This is a superb help feature, particularly for the novice. The pro will have a good time with this game, too.

On-screen graphics are equal to the best I've seen for ANY card game. The board on the right side runs hand totals to their new mark before filling in the peg area up to that point. There are three rows of 40 instead of the up-down 30/30 of normal cribbage boards; more like the continuous steep-sloped boards which are becoming popular.

But what kind of game does it play? It is a very good opponent that knows the rules perfectly. And you better not make an error in scoring or it will call "Muggins" on you and take the points; thus, the name "Cutthroat".

The only cheating I was able to do (and get away with) was to call a Go after a 25 when I had a three left in my hand. I got a point for just 54rd of 25. The computer called Go. I returned Go, though I could

From North Bay 99ers...

A REVIEW OF PULSADO PUBLISHER V 2.0

Live many reviews this is a review of Pulsado Publisher V 2.0. The program is a word processing tool that allows you to create and format documents. It features a variety of fonts, sizes, and styles, as well as the ability to insert graphics and tables. The interface is user-friendly and easy to navigate. The program is available for purchase on the Commodore 64 and Amiga platforms. It is a great tool for anyone who needs to create professional-looking documents.

Changes and additions: The program has been updated with several new features, including a new font set and improved printing options. The user interface has been redesigned to be more intuitive and easier to use. There are also several bug fixes and performance improvements. The program is now compatible with the Commodore 64 and Amiga platforms. It is a great addition to any word processing software collection.

Conclusion: Pulsado Publisher V 2.0 is a high-quality word processing program that offers a wide range of features and options. It is easy to use and produces professional-looking documents. It is a great choice for anyone who needs a word processing tool. The program is available for purchase on the Commodore 64 and Amiga platforms. It is a great addition to any word processing software collection.

Thanks for your time, Lotm. Stephen Andrews.....

have played. It played a four I played my three and got another point I certainly did not deserve. But I assume the author did not anticipate sneaky little buggers like myself trying to sneeze through a game instead of playing properly.

Except for the ability to cheat at the Go and the inability (on the computer's part) to note a Skunk or Lurch, this is a fantastic game in every way. NEW-AGE/99 rates this an A+! It is user-friendly! It is fast (though giving you time to make your crib-thru decisions!) It is nice to look at! It is cheap (\$7 gets you this and a couple other programs thrown in! \$10 gets you twice as many games, including AD's [CRIB]). The playing field is great. Your six cards are dealt at the bottom of the screen nearest you! The computer's opposite near the top. As you discard by pressing C,D,E,F,G, or H (letters assigned to your cards), the crib will be placed near your cards if you dealt or near the computer's if it dealt. You always know whose crib it is. The play is done card by card in the area between you and your opponent. Just like the real thing. Play scores (15, Run, Go, 31, Pairs, etc.) are all announced and scored during play. In addition to the multi-colored pegging "board" on the right, individual totals are kept in boxes next to each player. There is never a moment of confusion in this game.

At the conclusion of play, the hands and crib are spread out for analysis before scoring. Bear in mind that the computer's scoring is never wrong. All combos are sitting quietly in data statements just waiting for you to score incorrectly. If you do, you get zapped. You can never perform "hugging" on the machine, but sometimes its play isn't as sharp as your own under some odd play moments. It isn't as wise as some of the best cribbage players I've ever played. (My father, for instance, or my daughter Sue. Or an old Army buddy, Emil, from my days in Germany.) On the whole, though, Cutthroat is a worthy opponent, and, as cribbage cannot be played over the phone or by mail the way chess can, this is the next best thing to a human opponent. In some ways, better. It is 3:30 AM right now. I woke and danced The Insomniac's Waltz before settling at my computer. I popped on cribbage. Men the games. Felt good. Decided to write this column, which had been back-burnered for a couple weeks. Who else could I have gotten to play cribbage after 3 AM? And accept defeat so graciously?

Hey! Wait a minute! What if you don't play cribbage? Well, for one thing, you must have had a deprived childhood (and continually deprived adulthood) as cribbage is such fun! It's one of the few card games whose origins are known. It was created by the English gambler, soldier, poet (why so pale and wan, fair lover? Prithee, why so pale?) Sir John Suckling (1609-1642). It's come down to us across the centuries, virtually unchanged. The English still play the 5-card version, the Americans play the 6 (a variation of this century that caught on permanently in spite of considerable criticism by purists).

Anyway, if you don't play, get someone to teach you or get a Hoyler's and, with Cutthroat in front of you, teach yourself. This game makes it easy, particularly with the option of creating any hands you want to learn to score.

My fifth-grade students play it constantly and are even beginning to beat it almost as often as it beats them. Great learning tool. Finally, at game's end, there is no gloating. The screen fills with "YOUR VICTORY" or "MY VICTORY" without the obnoxious toots, plunks, and burps that usually accompany such announcements. Very professional. Now for those times when the urge of the cribbage addict comes upon you and no one is around to play, just pop a Foster's, turn on your Friendly TV, and go to it.

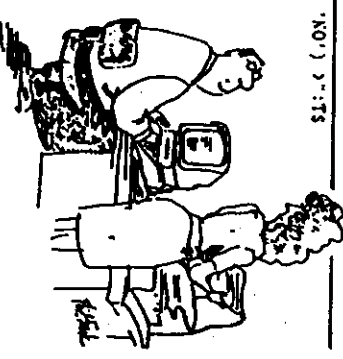
OGDEN T1 USER GROUP Page 6

```

50 REM *****ARC puts out of NR. General users may want to use GPL 5 to speed pro
100 REM *****VORP SEARCH GAME BY CHARLES STEINBERG---TI 99/4 EDITION *****
110 REM *****THE SCREEN PRINT HAS BEEN COMPRESSED SO IT WILL FIT ON LARGER PUZZL
120 REM *****IF YOU USE THE PRNG, WITH A *****
130 REM *****CHANG LINE 970 TO PRINT LS(1,J); ***** LINE 1060 TO PR
140 REM *****CALL CLEAR *****
150 REM *****PRINT TO YOU WISH COPY OF PUZZLE? (PRINTER ATTACHED) Y OR N)*****
160 REM *****CALL CLEAR *****
170 REM *****KANDIMIZL *****
180 REM *****DATA INITIATION *****
190 DIM WS(30,20)
200 DIM MS(30,20)
210 DIM MS(20)
220 DIM AS(320)
230 AS(1)=B
240 AS(2)=B
250 AS(3)=C
260 AS(4)=E
270 AS(5)=E
280 AS(6)=E
290 AS(7)=E
300 AS(8)=E
310 AS(9)=E
320 AS(10)=E
330 AS(11)=E
340 AS(12)=E
350 AS(13)=E
360 AS(14)=E
370 AS(15)=E
380 AS(16)=E
390 AS(17)=E
400 AS(18)=E
410 AS(19)=E
420 AS(20)=E
430 AS(21)=E
440 AS(22)=E
450 AS(23)=E
460 AS(24)=E
470 AS(25)=E
480 AS(26)=E
490 AS(27)=E
500 AS(28)=E
510 AS(29)=E
520 AS(30)=E
530 AS(31)=E
540 AS(32)=E
550 AS(33)=E
560 AS(34)=E
570 AS(35)=E
580 AS(36)=E
590 AS(37)=E
600 AS(38)=E
610 AS(39)=E
620 AS(40)=E
630 AS(41)=E
640 AS(42)=E
650 AS(43)=E
660 AS(44)=E
670 AS(45)=E
680 AS(46)=E
690 AS(47)=E
700 AS(48)=E
710 AS(49)=E
720 AS(50)=E
730 AS(51)=E
740 AS(52)=E
750 AS(53)=E
760 AS(54)=E
770 AS(55)=E
780 AS(56)=E
790 AS(57)=E
800 AS(58)=E
810 AS(59)=E
820 AS(60)=E
830 AS(61)=E
840 AS(62)=E
850 AS(63)=E
860 AS(64)=E
870 AS(65)=E
880 AS(66)=E
890 AS(67)=E
900 AS(68)=E
910 AS(69)=E
920 AS(70)=E
930 AS(71)=E
940 AS(72)=E
950 AS(73)=E
960 AS(74)=E
970 AS(75)=E
980 AS(76)=E
990 AS(77)=E
1000 AS(78)=E
1010 AS(79)=E
1020 AS(80)=E
1030 AS(81)=E
1040 AS(82)=E
1050 AS(83)=E
1060 AS(84)=E
1070 AS(85)=E
1080 AS(86)=E
1090 AS(87)=E
1100 AS(88)=E
1110 AS(89)=E
1120 AS(90)=E
1130 AS(91)=E
1140 AS(92)=E
1150 AS(93)=E
1160 AS(94)=E
1170 AS(95)=E
1180 AS(96)=E
1190 AS(97)=E
1200 AS(98)=E
1210 AS(99)=E
1220 AS(100)=E
1230 AS(101)=E
1240 AS(102)=E
1250 AS(103)=E
1260 AS(104)=E
1270 AS(105)=E
1280 AS(106)=E
1290 AS(107)=E
1300 AS(108)=E
1310 AS(109)=E
1320 AS(110)=E
1330 AS(111)=E
1340 AS(112)=E
1350 AS(113)=E
1360 AS(114)=E
1370 AS(115)=E
1380 AS(116)=E
1390 AS(117)=E
1400 AS(118)=E
1410 AS(119)=E
1420 AS(120)=E
1430 AS(121)=E
1440 AS(122)=E
1450 AS(123)=E
1460 AS(124)=E
1470 AS(125)=E
1480 AS(126)=E
1490 AS(127)=E
1500 AS(128)=E
1510 AS(129)=E
1520 AS(130)=E
1530 AS(131)=E
1540 AS(132)=E
1550 AS(133)=E
1560 AS(134)=E
1570 AS(135)=E
1580 AS(136)=E
1590 AS(137)=E
1600 AS(138)=E
1610 AS(139)=E
1620 AS(140)=E
1630 AS(141)=E
1640 AS(142)=E
1650 AS(143)=E
1660 AS(144)=E
1670 AS(145)=E
1680 AS(146)=E
1690 AS(147)=E
1700 AS(148)=E
1710 AS(149)=E
1720 AS(150)=E
1730 AS(151)=E
1740 AS(152)=E
1750 AS(153)=E

```

Fortunately the computer was not  
 as dumb as our records. It was immediately  
 abandoned by all the bugs in our programming.



```

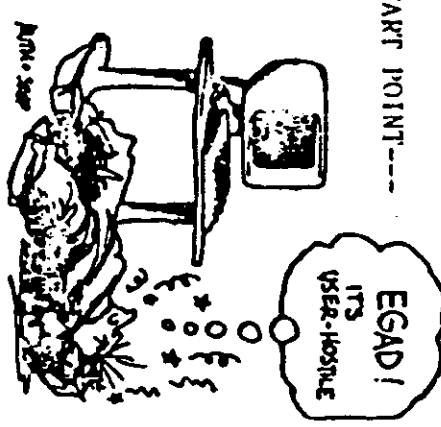
840 REM---TERMINATION POINT---
850 INPUT "WANT TO PLAY AGAIN(Y OR N)";BS
860 IF BS="Y" THEN 490
870 PRINT "THANKS FOR PLAYING. BYE";
880 STOP
890 REM---PUZZLE PRINT ROUTINE---
910 OPEN "1:PI0.D"
920 FOR I=1 TO 8
930 FOR J=1 TO 8
940 TONE F1,059463094:(INT(RND*24)+1)
950 CALL SOUND(50,TONE,50)
960 PRINT "1:";I;J;TONE;
970 IF C<>" " THEN 1030
980 IF C<>" " THEN 1030
990 IF C<>" " THEN 1030
1000 PRINT "1:";I;J;TONE;
1010 GOTO 1030
1020 PRINT "1:";I;J;TONE;
1030 NEXT J
1040 IF C<>" " THEN 1060
1050 NEXT I
1060 PRINT "1:";I;J;TONE;
1070 NEXT J
1080 NEXT I
1090 FOR I=1 TO 500
1100 IF C<>" " THEN 1130
1110 PRINT "1:";I;J;TONE;
1120 GOTO 1030
1130 RETURN
1140 REM---WORD REANNOX ROUTINE---
1150
1160
1170
1180
1190
1200
1210
1220
1230
1240
1250
1260
1270
1280
1290
1300
1310
1320
1330
1340
1350
1360
1370
1380
1390
1400
1410
1420
1430
1440
1450
1460
1470
1480
1490
1500
1510
1520
1530
1540
1550
1560
1570
1580
1590
1600
1610
1620
1630
1640
1650
1660
1670
1680
1690
1700
1710
1720
1730
1740
1750

```

```

1150 REM
1160 REM
1170 REM---LEN(WS(K))
1180 REM---RANDOM SELECTION OF DIRECTION---
1190 I=1+I
1200 IF T<100 THEN 1260
1210 PRINT "1 (GULN)T FIT THE WORDS IN - BOKRY, TRY WAIN"
1220 T=0
1230 GOTO 530
1240 P=1
1250 P=1
1260 P=-1
1270 IF RND<=.5 THEN 1290
1280 Q=1
1290 Q=-1
1300 IF RND<=.5 THEN 1320
1310 Q=1
1320 D=2
1330 IF Q<>1 THEN 1350
1340 P=0
1350 IF Q<>1 THEN 1370
1360 P=0
1370 IF RND<.75 THEN 1390
1380 D=1
1390 IF RND<.25 THEN 1410
1400 D=0
1410 REM---RANDOM SELECTION OF START POINT---
1420 C=0
1430 R=0
1440 IF D<>1 THEN 1460
1450 R=R-L0
1460 IF D<>0 THEN 1480
1470 C=C-L0
1480 IF D<=1 THEN 1510
1490 R=R-L0
1500 C=C-L0
1510 IF C<>C THEN 1530
1520 P=0
1530 IF R<>R THEN 1550
1540 P=0
1550 Z1=(R0*RND/100+.01)*100
1560 Z2=(C0*RND/100+.01)*100
1570 X1=INT(Z1+P*L0)
1580 X2=INT(Z2+P*L0)
1590 REM---ENTRY OF WORD IN THE PUZZLE---
1600 IF D=1 THEN 1620
1610 IF D=0 THEN 1720
1620 FOR I=1 TO L0
1630 IF L$(X2+(I-1)*Q0,X1+(I-1)*Q)="" THEN 1650
1640 IF L$(X2+(I-1)*Q0,X1+(I-1)*Q)<>SEGS(WS(K),I,1) THEN 1180
1650 NEXT I
1660 T=0
1670 FOR I=1 TO L0-1
1680 L$(X2+1*Q0,X1+1*Q)=SEGS(WS(K),I+1,1)
1690 NEXT I
1700 L$(X2,X1)=SEGS(WS(K),I,1)
1710 GOTO 1910
1720 FOR I=1 TO L0
1730 IF L$(X2+(I-1)*Q0,X1)="" THEN 1750
1740 IF L$(X2+(I-1)*Q0,X1)<>SEGS(WS(K),I,1) THEN 1180
1750 NEXT I

```



```

1760 T=0
1770 FOR I=1 TO 10-1
1780 L$(V2+I*Q0,X1)=SREG(W$(K),I+1,I)
1790 NEXT I
1800 L$(V2,X1)=SREG(W$(K),I,I)
1810 GOTO 1910
1820 FOR I=1 TO 10
1830 IF L$(V2,X1+(I-1)*Q)="" THEN 1180
1840 IF L$(V2,X1+(I-1)*Q) <> SREG(W$(K),I,I) THEN 1180
1850 NEXT I
1860 T=0
1870 FOR I=1 TO 10-1
1880 L$(V2,X1+I*Q)=SREG(W$(K),I+1,I)
1890 NEXT I
1900 L$(V2,X1)=SREG(W$(K),I,I)
1910 RETURN
1920 REM---FILL. OF REMAINING POSITIONS---
1930 FOR I=1 TO 6
1940 FOR J=1 TO R
1950 IF L$(I,J) <> "" THEN 1980
1960 Z1=(RND*.26+.01)*100
1970 L$(I,J)=AS(INT(Z1))
1980 NEXT J
1990 NEXT I
2000 REM -JUMBLE SORT TO ARRANGE WORD LIST ALPHABETICALLY-
2010 FOR I=1 TO N-1
2020 IF W$(I+1) > W$(I) THEN 2070
2030 LET W$(I+1)=W$(I)
2040 LET W$(I)=W$(I+1)
2050 LET W$(I)=W$(I)
2060 GOTO 2010
2070 NEXT I
2080 REM-PRINT TITLE TO PRINTER AND WORD LIST-
2090 IF ($ <> "Y") THEN 2160
2100 OPEN #1:"P10"
2110 IF "$="NO" THEN 2140
2120 PRINT #1: " "
2130 PRINT #1: " >>" ; TS ; "<<"
2140 PRINT #1: " "
2150 PRINT #1: " *WORD LIST*"
2160 PRINT
2170 CALL CLEAR
2180 PRINT "WORD LIST":
2190 FOR K=1 TO N
2200 CALL SOUND(200,K*220,20,2300/K,20)
2210 PRINT " " ; W$(K) ;
2220 IF ($ <> "Y") THEN 2240
2230 PRINT #1: W$(K)
2240 NEXT K
2250 IF ($ <> "Y") THEN 2280
2260 PRINT #1:
2270 CLOSE #1
2280 RETURN
2290 END

```

EXPANDING YOUR SYSTEM  
Adding a disk drive, what to expect and  
what to look for.

by John Kolan

I think it is safe to say that anyone who is serious about using his computer should seriously consider acquiring a disk drive.

A disk drive and controller are more important than memory expansions, RAM disks, printers or color monitors. The only piece of equipment that might be more important than a disk controller/disk drive system is a Peripheral Expansion Box, though you can have a disk drive system without a PEB by using a CorComp Micro Expansion System or stand-alone sidescar disk controller.

For the purposes of this article, we'll look at three system variations: a PEB with controller and disk drive; and briefly a MES with a disk drive. The last of these is not recommended, but it is better than relying on a cassette recorder for data storage.

ENTRY LEVEL CASSETTE SYSTEM

Cassette storage of data is an entry level system that allows buyers of the TI console to save and load relatively short BASIC and Extended BASIC programs. It was not designed to utilize memory expansions, assembly language programs or other powerful tools. This isn't a bad arrangement as long as you are satisfied with using cartridge based software.

However the best software isn't available on cartridges. In fact, there are devices, which we'll get into in a subsequent installment, that allow you to dump the contents of most cartridges to disk and load and run them without the cartridge.

Anyone who has upgraded from a cassette to a disk system probably remembers the first time he used the disk drive. I know I do. The first time I accessed the drive I knew I was into something completely different and wonderful. It was akin to the feeling I had the first night I took the TI console out of the box and plugged it into a black and white TV on the kitchen table. It was like Christmas to a nine-year-old. Only I wasn't nine. With the addition of a disk system, you are catapulted into a new realm of computing. Suddenly you are able to load and run programs in seconds, operations which took minutes with a cassette recorder. You can load and save these programs by name, too, instead of keeping track of their locations on a tape counter. No longer are you limited to viewing computer operations in a sequential manner. Yes, owning a disk drive system can change your computing life. Now to the details.

DISK CONTROLLERS

Selecting a disk controller is relatively straightforward. If you have a PEB, you'll need a disk controller card. The card fits into the PEB. Disk controller cards have been

manufactured by a number of companies, including Texas Instruments, Nyrac Inc. and CorComp Inc. Disk controllers are used to communicate between the computer CPU and the disk drive. The controller makes it possible to format disks--prepare them to hold data--and to copy, delete, protect or unprotect files and programs. These functions are done through disk manager software which is provided with the disk controller. Each of the three above mentioned controller cards has unique characteristics. Examine the chart below:

**DISK CONTROLLER CAPABILITIES**

Disk format	CorComp	Nyrac	TI
single-sided	Y	Y	Y
double-sided	Y	Y	Y
single-density	Y	Y	Y
double-density	Y	Y	N
quad-density	N	Ye	N

\*requires special EPROM

Density as used here refers to the compactness of the information stored on a disk. A double-density disk holds twice as much data as a single-density disk. Similarly, a double-sided disk holds twice as much data as a single-sided disk. See the chart below:

**CAPACITY OF DIFFERENT FORMATS**

Disk format	Capacity
SSSD	90 Kilobytes
SSDD	180 Kilobytes
SSDD	180 Kilobytes
DSDD	360 Kilobytes

You should notice that a double-sided, double-density drive lets you store four times the data as a single-sided, single-density drive. All other things being equal, you'd be better off with a DSDD drive than a SSDD drive.

**CONTROLLER DETERMINES FORMAT**

However, even if you have a DSDD drive, you may not be able to format DSDD disks. This depends on the capabilities of the disk controller. The TI disk controller, for example, doesn't support double-density formats. Incidentally, although manufacturers label their diskettes as being suitable for double-sided, double density use, they are equally suitable for use as single-sided, single-density formatting.

Both the CorComp and Nyrac floppy disk controllers (including the Nyrac Hard and Floppy Disk Controller) support single-sided single density through double-sided double-density. Here is a chart showing the formatting

characteristics of the three controllers:

Format	CorComp	Nyrac	TI
SSSD	360	360	360
SSDD	720	640	N/A
DSDD	1440	640	720
DSDD	1440	1280	N/A

A sector as used here refers to the subsections of a track that hold the stored data. It is the smallest addressable space on disk media. A track consists of a collection of sectors. In the case of the SSSD diskette, there are nine sectors per track and forty tracks for 720 sectors. A DSDD disk would have 18 sectors multiplied by forty tracks for each of its two sides to equal 1440 sectors.

The Nyrac controller uses nine sectors for SSDD and eight sectors for another format. It uses eight-sector tracks because Nyrac regards eight sectors to be a better format than nine.

The upshot of this is that beyond SSSD, disks formatted on these controllers are not compatible with each other. In other words, a disk formatted using eight sectors per track cannot be read by a controller that looks for nine sectors per track. This is one reason why the bulk of TI software is distributed on SSDD disks to maintain compatibility with all controllers.

Which is better: 8 or 9 sectors per track? I don't think it matters to the majority of users. Some people like CorComp products, others like Nyrac, and the rest don't care one way or the other. Both are more flexible than the TI controller. Incidentally, both the CorComp and Nyrac floppy controllers will handle up to four floppy disk drives, and the TI will handle up to three.

Once you've decided on a controller, the next step is to obtain a disk drive. This is the easy part. Virtually any 5.25-inch floppy drive that will work on a PC will work on a TI. This means that disk drives meant for Commodore 64 or Atari or Apple computers won't work. But those that are installed in PC clones will. They are available from a variety of sources, including advertisers in this magazine.

Assuming you've got a PEB, you will be installing a drive in the PEB itself. Since the PEB has its own power supply all you will need is the drive. But should you get a full-height drive or should you get a half-height driver?

A full-height drive is twice the width of a half-height drive. The PEB drive slot is designed to accept a full-height drive. However, it will also hold two half-height drives. And two drives are better than one. But, if you try to install two half height drives in the PEB you have to modify the power connector. If you aren't handy around electrical devices, you may want to purchase a kit designed to make this project simple. Tax-Comp sells an installation kit for two half-height drives for \$29.95. Those who are more adventurous or knowledgeable can proceed on their own.

Let's say for now that you want to install a single drive in the PEB. I recommend that whatever you get that it be capable of handling DSDD disks. These drives are available virtually every where since these are the drives that are used in PCs. Even if you use a TI controller card, which can't format DSDD, it can still use the drive for DSDD SSSD. Generally, the DSDD drives are better built than the older DSDD drives.

#### MAKE SURE YOU'VE GOT CABLES

Once you've got a drive, you'll need to have cables. In most cases, your disk controller card should come with a short ribbon cable to connect a drive to the PEB, a second longer ribbon cable to connect the controller to external drives, and a small wafer board adapter used to connect external drives. (If you but a used controller, make certain that it includes the two cables, the adapter board and a complete manual. If it doesn't include the cables you will definitely be inconvenienced in obtaining what you need, especially if you don't live in a relatively large city with large electronic supply businesses.)

The short cables is used to connect the internal PEB drive to the disk controller. One end plugs into the disk drive and the other plugs into the controller. A four-pin molex connector is used for connecting the disk drive to the PEB power supply. Once the cables are connected, you seat the drive in its slot and fasten it with machine screws.

External floppy drives (drives that aren't installed in the PEB) require a power supply and disk drive enclosure. You will also need to set some dual inline pins (DIP) according to the manufacturer's instructions. You'll also need the cable and adapter board that comes with the controller card.

The manuals that come with any of the above mentioned controllers are quite thorough about installation procedures, and I won't try to duplicate them here. The NES comes with the CorComp disk controller installed. You plug a cable from the NES to the drive and off you go. However, the drive must have its own enclosure and power supply since the NES doesn't have a power supply. Like the card version of the CorComp controller, the NES version also handles up to four floppies.

#### DISK DRIVE WITH A SIDECAR CONTROLLER

With a sidecar controller, you are pretty much on your own. TI manufactured these devices prior to developing the PEB. The controller plugs into the right side of the console and disk drive is plugged into the sidecar controller. Like the NES, the drive you use must have its own enclosure and power supply. The TI controller supported only SSSD drives. Several manufacturers produced these types of devices but none has been produced for several years. I you find one, it will probably be used. It is unlikely that you will ever find anyone to support these third-party controllers. I do not recommend them since they could easily prove to be more trouble than they are worth, especially for novices. Those who currently use these devices more than likely purchased them when they were still being produced and could obtain

support from the manufacturer or vendor.

#### WHAT'S IT GOING TO COST?

You can figure on spending about \$150 for a new floppy disk controller (including cables and manual). Cost of floppy drive without power supply for installation in the PEB will vary depending on whether you go full-height or half-height. For a new full-height drive, expect to pay \$60 or less, and for a new half-height drive expect to pay \$80 or less. For external drives, the cost of a full or half-height drive with enclosure and power supply should set you back about \$120.

If I were going to install two drives, I'd give serious considerations to obtaining two half-height drives and an installation kit for the PEB since this would avoid the additional cost to purchase and external enclosure and power supply.

Avoid the temptation of buying the cheapest drive you can find. In most cases, you won't actually be able to examine the drive before purchasing it so either select a well-known brand or buy from a vendor that you are familiar with or who has been in business for years. Established business get established because the satisfy their customers. Prices of used equipment vary considerably. Good places to look for newspapers and other computer publications. However, most other magazines charge so much for classified ads that only commercial dealers can afford them. Also, check out pawn shops, Goodwill stores, flea markets, computer fairs and other places that deal in second hand goods.

The main word of advice I have on purchasing used equipment is to make sure that it works before you pay for it. A used disk drive for \$30 may look like a great bargain. But if it needs to be realigned - a common problem with older drives - it could easily cost you another \$50 to get it to work. Specially your \$30 drive costs more than a new one.

You may have noticed that the main article didn't have much to say about Myarc's Hard and Floppy Disk Controller card. That's because this device is in a class by itself, both in terms of price and capabilities.

The device costs about \$250-\$300 and allows you to control up to four floppies and three hard drives in a 99/4e. With the additions of an 80-track EPROM, it can also use quad-density, 3.5-inch drives. Disks formatted on these drives hold up to 720 Kibibytes of data, twice what a DSDD 5.25-inch disk will hold.

I recommend the HPDC to anyone who eventually wants to upgrade to a hard disk system. It is the only hard disk controller available for the TI, other than the Myarc's old Winchester hard disk (reviewed in the March 1985 issue of MICROpendium).

The HPDC was reviewed in the September 1988 issue of MICROpendium.



I am reserving the copyright on this material, but I will allow the copying of this material by anyone under the following conditions: (1) It must be copied in its entirety with no changes. (2) It is to be printed, stored on a hard disk, or otherwise made available for your personal use only. (3) The printed copy must be for your own use only. (4) It may not be copied, stored, or otherwise made available for the copying or transfer of this material. In other words, they can use it to their instructors and you can give a copy to your friend as long as it's free.

When I returned to a station of total insanity with Martin A. I discovered a bunch of things that I had changed or deleted over the months without being told. A few days ago I thought I'd try to get the system and jump back into writing the Tutorial. What a surprise! The system decided it didn't like me. I gave it all kinds of different commands, but it wouldn't do what I wanted it to do. At the same time I was having all these problems I was hearing through my old notes and letters and I found that many other people had the same problem at one time or another. I cut this story short. I came to the conclusion that any of my explanations of T-base functions were inadequate. They were only words if you already knew what you were doing. While my notes were giving me problems, one to a lack of use, I had read a little bit of what I had to write to provide myself with a readable and interpretable format. I can imagine that there are 100 of people out there trying to learn it, this way is not the way. I don't understand a word of this junk. So I'm going to try to reorganize my thoughts and try again. It may not be any better than before, but this time I will try to include some of the notes I go through to accomplish a task in hopes that you will pick up some ideas along the way.

For this month's article I decided that I wanted to start writing with more notes, and possibly, with the use of more lines over than one page at a time. I decided to work on the utility bills for 1987 and a checkbook-type no. I do not see track of my bills or my checkbook on my computer, so I thought I could do this project as a beginner. I expect to do a lot of mistakes in planning and will probably change the file several times in the next couple months. I started with a text and paper, as usual, and listed some utilities with

dates and checknumbers. I came up with the bills by you see at the bottom of this page. I do not have all the bills, but I'll be sure and mail them to you with the printer and checknumbers. This will give me something to play with. I created this on the specifications you see in the bottom right corner of this page. When I typed it I wanted to enter data, I answered YES, and just held the ENTER key until I had finished blank records. Then I pressed F10 to get out of MODEM and typed G17 (EX). Being able to enter your data into that field is helpful, because you can page up to see what's in the previous record. If you use this 1989, you will remember to use F10 to record the data as you enter it. Unfortunately there are some things you will have to hard type. I filled in a bunch of data, payments and checknumbers in what I hope is a logical manner for me.

```

3. NUMBER CHECK DB          CHKNUM/C
LOCAL NUMBER N 4
REPLACE NUMBER WITH 245
USE CHKBK
WHILE (NUMBER<330)
APPEND BLANK
REPLACE CHKNO WITH NUMBER
REPLACE NUMBER WITH NUMBER + 1
ENDWHILE
CLOSE ALL
RETURN Copyright Martin A. Smolow 1989

```

```

3. MOVE NUMBERS TO CHKBK FLCKB/C
SELECT 2
USE CHKBK
SELECT 1
USE BILLS
WHILE .NOT. (EOF)
SELECT 2
FIND 1.PCHK
IF .NOT. (EOF)
REPLACE 2.PAYDEBIT WITH 1.PHONE
REPLACE REMARKS WITH "Phone "
ENDIF
SELECT 1
MOVE
ENDWHILE
CLOSE ALL
RETURN Copyright Martin A. Smolow 1989

```

```

*****
e Continued e
e Next Page e
*****
CREATED 11/16/89 CHANGED 11/19/89
FIELD DESCRIPTION TYPE WIDTH DEC
1. MONTH          B     008      02
2. PHONE          B     007      02
3. 545           B     007      02
4. ELECTRIC      B     007      02
5. WATER        B     007      02
6. WASTE        B     007      02
7. TAX          B     007      02
8. PCHK         C     004
9. PCHK         C     004
10. PCHK        C     004
11. PCHK        C     004
12. PCHK        C     004
13. PCHK        C     004
14. REMARKS     C     440

```

```

*****
e Continued e
e Next Page e
*****
CREATED 11/16/89 CHANGED 11/19/89
FIELD DESCRIPTION TYPE WIDTH DEC
1. MONTH          B     008      02
2. PHONE          B     007      02
3. 545           B     007      02
4. ELECTRIC      B     007      02
5. WATER        B     007      02
6. WASTE        B     007      02
7. TAX          B     007      02
8. PCHK         C     004
9. PCHK         C     004
10. PCHK        C     004
11. PCHK        C     004
12. PCHK        C     004
13. PCHK        C     004
14. REMARKS     C     440

```

After I filled bills, I decided that a checkbook-type bill would also be interesting. I then proceeded to CREATE MODEM. The structure of bills is located in the bottom left corner of 12.1.2. After the CREATE screen, when I held the ENTER key until I had finished blank records, then I held the F10 key, first of all I held to enter data, and second, this is a pre-reading tutorial, so I waited 110 to save the data for me. I used MODEM (CHANGED) to skip up the bills to the middle of the previous page. The top line of this CF is a constant 110. The next line (LOCAL NUMBER N 4) tells 110 to allocate space that I will use to store a NUMBER, with a length of 4 and label the space NUMBER. You should create the LOCALS you need at the beginning of the CF. If they are all together and up front, they are easier to create when you're having problems. Note it is easier to create locals with a WHILE loop. REPLACE MODEM WITH 345 puts the number 345 into MODEM. I chose 345 because it seemed like a good place to start numbering the checkbook. If you wanted the first check number to be 1, you could use 1 instead of 345. MODEM opens the checkbook no. WHILE (NUMBER<330) is a loop that will go around and around as long as the value in NUMBER is less than 330. As long as NUMBER is less than 330, all the commands you have placed between the WHILE and ENDWHILE will be carried out, if possible. The first command, APPEND BLANK, adds one whole empty record on the end of CHKBK. Now that we have some space, let's store some data there. The next line takes our NUMBER, which is 345, and places a copy of it into our NUMBER, which is 345, and places a copy of it into CHECK. CHECK is an abbreviation for check number. The next line takes our number out of MODEM (345), adds 1 to it (346), and sticks that back into MODEM. It's worth this line the check number will not increase by 1, and we would never reach 330 to get out of this loop. So this little CF takes the 345 through 331. A short version of CHECK is on the right side of this page. I added the deposit items at the top by hand, the check numbers were put in with CHECK and the other data was saved to CHECK by modified versions of the CF named FICENCY, short for Fill Check Book Phases. At this point there are only blank spaces in both of our files, but that doesn't matter now. I want to press on with more pre-reading files and also save our constants.

```

*****
e Continued e
e Next Page e
*****
CREATED 11/18/89 CHANGED 11/20/89
FIELD DESCRIPTION TYPE WIDTH DEC
1. CHECKNO      C     004
2. DATE        B     008      02
3. PAY/REMIT  B     009      02
4. NET         B     003
5. FEE/CHGS   B     008      02
6. REP/DEBIT  B     009      02
7. BALANCE    B     010      02
8. REMARKS     C     440

```

```

*****
e Continued e
e Next Page e
*****
CREATED 11/18/89 CHANGED 11/20/89
FIELD DESCRIPTION TYPE WIDTH DEC
1. CHECKNO      C     004
2. DATE        B     008      02
3. PAY/REMIT  B     009      02
4. NET         B     003
5. FEE/CHGS   B     008      02
6. REP/DEBIT  B     009      02
7. BALANCE    B     010      02
8. REMARKS     C     440

```

```

*****
e Continued e
e Next Page e
*****
CREATED 11/18/89 CHANGED 11/20/89
FIELD DESCRIPTION TYPE WIDTH DEC
1. CHECKNO      C     004
2. DATE        B     008      02
3. PAY/REMIT  B     009      02
4. NET         B     003
5. FEE/CHGS   B     008      02
6. REP/DEBIT  B     009      02
7. BALANCE    B     010      02
8. REMARKS     C     440

```

After I filled bills, I decided that a checkbook-type bill would also be interesting. I then proceeded to CREATE MODEM. The structure of bills is located in the bottom left corner of 12.1.2. After the CREATE screen, when I held the ENTER key until I had finished blank records, then I held the F10 key, first of all I held to enter data, and second, this is a pre-reading tutorial, so I waited 110 to save the data for me. I used MODEM (CHANGED) to skip up the bills to the middle of the previous page. The top line of this CF is a constant 110. The next line (LOCAL NUMBER N 4) tells 110 to allocate space that I will use to store a NUMBER, with a length of 4 and label the space NUMBER. You should create the LOCALS you need at the beginning of the CF. If they are all together and up front, they are easier to create when you're having problems. Note it is easier to create locals with a WHILE loop. REPLACE MODEM WITH 345 puts the number 345 into MODEM. I chose 345 because it seemed like a good place to start numbering the checkbook. If you wanted the first check number to be 1, you could use 1 instead of 345. MODEM opens the checkbook no. WHILE (NUMBER<330) is a loop that will go around and around as long as the value in NUMBER is less than 330. As long as NUMBER is less than 330, all the commands you have placed between the WHILE and ENDWHILE will be carried out, if possible. The first command, APPEND BLANK, adds one whole empty record on the end of CHKBK. Now that we have some space, let's store some data there. The next line takes our NUMBER, which is 345, and places a copy of it into our NUMBER, which is 345, and places a copy of it into CHECK. CHECK is an abbreviation for check number. The next line takes our number out of MODEM (345), adds 1 to it (346), and sticks that back into MODEM. It's worth this line the check number will not increase by 1, and we would never reach 330 to get out of this loop. So this little CF takes the 345 through 331. A short version of CHECK is on the right side of this page. I added the deposit items at the top by hand, the check numbers were put in with CHECK and the other data was saved to CHECK by modified versions of the CF named FICENCY, short for Fill Check Book Phases. At this point there are only blank spaces in both of our files, but that doesn't matter now. I want to press on with more pre-reading files and also save our constants.

```

*****
e Continued e
e Next Page e
*****
CREATED 11/18/89 CHANGED 11/20/89
FIELD DESCRIPTION TYPE WIDTH DEC
1. CHECKNO      C     004
2. DATE        B     008      02
3. PAY/REMIT  B     009      02
4. NET         B     003
5. FEE/CHGS   B     008      02
6. REP/DEBIT  B     009      02
7. BALANCE    B     010      02
8. REMARKS     C     440

```

```

*****
e Continued e
e Next Page e
*****
CREATED 11/18/89 CHANGED 11/20/89
FIELD DESCRIPTION TYPE WIDTH DEC
1. CHECKNO      C     004
2. DATE        B     008      02
3. PAY/REMIT  B     009      02
4. NET         B     003
5. FEE/CHGS   B     008      02
6. REP/DEBIT  B     009      02
7. BALANCE    B     010      02
8. REMARKS     C     440

```

```

*****
e Continued e
e Next Page e
*****
CREATED 11/18/89 CHANGED 11/20/89
FIELD DESCRIPTION TYPE WIDTH DEC
1. CHECKNO      C     004
2. DATE        B     008      02
3. PAY/REMIT  B     009      02
4. NET         B     003
5. FEE/CHGS   B     008      02
6. REP/DEBIT  B     009      02
7. BALANCE    B     010      02
8. REMARKS     C     440

```

The CR used FICR2 is still not very powerful. If you can give the logic in its functions you will see how it is working out the use of 11-Base. All right. The top line of FICR2 is for constants only. SELECT 2 tells 118 to switch to workspace 2 and open (USE) CHECK in that workspace. Next 118 is told to go to workspace 1 (SELECT 1) and open 118 in that area. We now have 2 workspace. CHECK in workspace 2 and 118 in workspace 1. You will have trouble remembering where things are found, but 118 will not when you forget and select 118, 118 will ask you what the heck you are doing. SELECT 1, USE 118 and WHILE .NOT. (END) will be lined together in your mind. We want to first through 118 and record at a time and do everything inside the WHILE loop until the END OF FILE (EOF) of 118 is reached. After at this point we are automatically switching to record number one in 118. SELECT 2 tells 118 to forget about 118 (in 1) and start looking at CHECK which is selected in workspace 2. 118 will be looking at CHECK to see the contents in that workspace. FIND 1, CHECK is a very powerful (usable) statement. It says, go to work area 1 and find the value under CHECK. 118 doesn't care which record it is currently pointing at, that is up to you to remember. In 1 just before we are looking at record one. Therefore, you will find that. That is the state thing 118 will come up with. CHECK will be SPICED ON CHECK. We are now looking at CHECK (in 2), and the 2nd record tells 118 to look at whatever field is marked. 118 is looking through the CHECK field for the number 249, which it got from 118. If it finds a match for 249, it will not do anything, it just stops there with the record of the first 118 record with the record in the record 24. This alignment will allow us to save data from one record to a corresponding record. 118 searches the entire file and can't find a match, it will stop at the END OF FILE and there will be an EOF from 118. I don't tell you this, you just test for it. So the logic is this, if on hit the EOF then we didn't get a match and that means we don't want to do anything. However, if we did not hit the EOF, or if .NOT. (END), then we found a match and we can save data from 118 to CHECK. In this case we REPLACE 2, MITE (END), REPLACE 2, 249, REPLACE 118 1, MITE and REPLACE (END) WITH "Phone ". After I have followed plan that requires equal length of the first file in the RECORDS field. Also that records have a length of 40, but I changed out of the record length for this problem. As we pass out of the IF statement at EOF, 118 is waiting in area 2, the next switch from the ENDFILE is remembered its job is to loop 118 back to the WHILE. The WHILE then tests to see if we have hit the EOF. If 1 goes well, FICR2 will switch up one of the CHECK field in the WHILE and the WHILE would loop for the (END) in CHECK. If doesn't care or know which it's working on. You must realize these things and tell 118 exactly what you want it to do. If I go well, FICR2 will switch up one of the CHECK field in the WHILE with a CHECK field in CHECK and copy the data, amount and then " into CHECK. This CR only covers the Phone data. I have a data field in area 2 and 11-Base in area 1, so I typed FICR2, FICR2/C, FICR2/C, FICR2/C, then COPY FICR2/C, FICR2/C, FICR2/C, etc. until I had a copy with the same changes. Then I read FICR2.

Changed to edit each CR. I changed the first letter in the FIND statement field (FICR) to represent bin, or Electric, or Snow, etc. I also changed the last field name in the middle REPLACE statement (PHONE) to SNOW, or ELECTRIC, etc. Last I changed the RECORDS only from "Phone " to "Snow " or "Electric", etc. To correspond to the CR I was working on. You need a different CR to handle all the CHECK fields in 118. It's and by the time you correct everything you'll have a lot of experience working with CR.

Here's something new. PRINT ALL FROM CHECK. The part that I haven't discussed before is (FROM CHECK). This is basically a filter. It filters out everything you don't want printed. In this case I have opened CHECK (USE CHECK), and I would like a printout of all of Bopposit. Since I have the Bopposit do not have check numbers I told 118 to print everything that didn't have a check number. (ISPLAN CHECK, MITE FROM MITE(94/91/97) would display the original (NAME, MITE FROM MITE(94/91/97) would print only records dated March, (03). The (FROM) command is a very powerful tool.

```

CLOSE ALL
USE CHECK
PRINT (D-F-C), (4)
PRINT ALL (FOR (DATE>02/29/89-).AND.(
    (DATE<=04/00/89-))
    )
FORST1

```

Here is a tip that is so valuable you should all give an money, but you can have the tip for free. Try to never input commands at the command line. Always create a Grand File (GF), an editor has said or insignificant. I use MODIFY COMMAND (FORST1) to create FORST1 which tells the state line you see in my text just prior to that. I tried the command five or five times before I got the result I wanted. By the time I changed the line three times I already forgot my old mistakes and started making them over again. For this problem I use another trick. I save the old command line in the CR, but I place an asterisk at the beginning of the line so 118 will ignore it. Then I retype the new line below that for another try. At a later date I can type LIST BOP, FORST1/C, to get a printout of FORST1 and see exactly which attempt worked and which didn't. You do not have to save anything except in these little CRs, but you should always open and copy commands to see the name of the BOP field with the command and save it for later. It will close the database for you. Many times I have gotten discouraged and changed disks or reloaded the disk and turned the system off while a BOP was still open. There are also many other reasons that will kill that will stop your database if you leave it open while you performing 118 high disk access commands such as COPY, because I don't remember which commands are dangerous, I try to keep all of the closed editors I am performing a specific task and once again I standstill close them. You will also notice that I have created notes inside the CR such as, This works, but not, This don't work, etc. One final comment. It took me a month to complete this tutorial from the first time to the finished product.

OGDEN T1 USER GROUP Page 10  
 Continued Next Month.

# Calendar Programs

About this line of year it's continue any for programmers to come out with various programs that produce calendars. Of course it is necessary to incorporate the correct algorithms for a correct calendar, but any program suffer from insufficient programming. A calendar program can really be quite simple, but an understanding of how our present day calendar came about is necessary to produce an efficient program.

The history of the development of the calendar is an interesting subject. A calendar is basically just an enumeration of the days in a year, or of the number of rotations of the earth during one revolution around the sun. Our present day calendar is based on the tropical year which is the time between two consecutive occurrences of the "Vernal Equinox", the time when the sun's apparent motion crosses the equator. The length of the tropical year is approximately 365.2422 mean solar days. We will refer to this value later.

In order to properly understand our present day calendar we need to introduce our early civilizations and how in their days they celebrated the seasons. They noted that the sun set and rose at different places on the horizon and they decided to mark the beginning of the year by the passing and reappearing of the sun for the length of the day. They an early crude calendar was devised. The results of our present day calendar go back some 7000 years. The Egyptian first distinguished the length of the year as 360 days and then calendar later upgraded it to 365 days. With this knowledge they devised a calendar of lunar thirty-day months supplemented by five consecutive year-end holidays. That's bad! That's how they realized that

The year was more nearly 365 1/4 days long and in 525 B.C. they provided for a leap year every fourth year. This calendar however was not widely accepted. Assembling the Roman were struggling with a lunar calendar of twelve alternating 29 and 30 day months. The moon cycle is 29.5 days. But twelve lunar months total only 354 days so they would be necessary to throw in an extra month about every two or three years to keep up.

In 46 B.C. Julius Caesar reformed the Roman Calendar. He made it longer, not of the same and changed the length of the months to total 365 days including the leap year year. He also reformed the time of the vernal equinox to the ancient date of March 21. Eventually this became the Julian Calendar. The next change was the division of the month into the Christian 7-day week. This added a further complication to the calendar since the year is not divisible by seven. Now the Julian Calendar of 365.25 days differs from the actual length of the tropical year of 365.2422 days by about one day every 129 years. This caused the calendar date of the vernal equinox to advance from March 25 to March 16 by the 15th century century. If this were to continue we would have spring in December and winter in August.

In 1582 Pope Gregory XIII proceeded to correct the errors of the Julian Calendar. His reason was that the date of Easter was prescribed in A.D. 325 to be the first Sunday on or following the day of the first full moon following the vernal equinox. He realized that century years would not be counted as leap years unless they were also divisible by four hundred. This was the effect of adding the average length of the calendar year to the Gregorian calendar. The Gregorian calendar for any year after 1582, with any modifications it could be referred to print in various formats. Set your printer to line 20 and the print style in line 37.

OGDEN T1 USER GROUP has provided a line-by-line description of his program. In any case this will be helpful, see the follow:

```

100 CR: CALENDAR PROGRAM
101 BY: Martin Johnson
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

