



# The Ottawa T.I.99/4A Users' Group



VOLUME 8 NUMBER 3.....March 1989



DON'T FORGET THE MEETING -- March 7, 1989

P.O. BOX 2144, STATION D, OTTAWA  
\*\*\* ONTARIO, CANADA K1P 5W3 \*\*\*

## COMING EVENTS

|                                       |                            |                                                                                                 |
|---------------------------------------|----------------------------|-------------------------------------------------------------------------------------------------|
| Beginners' Assembly<br>Language Class | March 4, 1989<br>1:00 p.m. | Merivale High School<br>For more information<br>Contact Bill Sponchia.                          |
| March Meeting:                        | March 7, 1989<br>7:30 p.m. | Merivale High School                                                                            |
| Cassette users'<br>Workshop:          | March 11, 1989             | Merivale High School<br>Contact Bill Sponchia or<br>Jack McAllister for<br>further information. |
| Beginners' Assembly<br>Language Class | April 1, 1989<br>1:00 p.m. | Merivale High School<br>For more information<br>Contact Bill Sponchia.                          |
| April Meeting:                        | April 4, 1989<br>7:30 p.m. | Merivale High School                                                                            |
| 4th Annual TI-FEST                    | April 29, 1989             | Merivale High School<br>Contact Jane Laflamme<br>for details or to<br>volunteer your help.      |
| Newsletter Deadline:                  | March 15, 1989             | (or any time before that!)                                                                      |

## EDITOR'S NOTES from Ruth O'Neill

I'm encouraged. I really am. This month, two people handed me articles unexpectedly. For me at least, these people are new writers, so that enriches the newsletter even more. Let's show a very warm welcome to Art Green and Marcelle Gibson. They deserve it. Also, please don't forget our regulars, who are what make it possible for me to believe that there will be a newsletter each month. If you enjoy an article in the newsletter, tell the author! They will appreciate immensely.

This doesn't mean, however, that I'm not STILL looking for new people. I am-- and I still insist that you don't have to consider yourself a "great writer" to contribute here. If you have ANYTHING you would like to share, let me know.

Please note in the Coming Events schedule above, the cassette/console users workshop is on March 11 instead of the 18th. Because of the spring break, the school was unavailable for the third Saturday. If you do not plan to attend yourself, please make sure that others around you are aware of the opportunity. Jack McAllister is putting a great deal of effort into making this an extremely worthwhile session for ANY TI user who does not have a disk drive system. This means member and non-member alike. This workshop is open to all, so if you have a neighbour or co-worker with a TI who doesn't quite know what to do with it, send them to Jack. I'm sure he'll have some interesting information for them.

By the way, if you have any ideas or wish lists for these Saturday seminars, contact Bill Sponchia. These are for YOU, and the only way we can offer what you want is if you tell us what you want.

If you have any time at all to spare, talk to Jane about helping out for the Fest. She'll appreciate it!

The President's Two Cents' Worth  
by Jane Laflamme

We had a turn-out of about 30 people for the KOA-Didak plant tour on February 21st! (It was a stormy night, if you remember, so it was great to see people who braved the elements to get there.) We were taken on a tour of the 3 1/2" floppy plant. Things that impressed me personally: It was almost totally automated (99.9%); the precision required for manufacturing the components; the equipment itself required to do that; quality control; and, the pride that was shown by those who took us around. We would like to thank Mrs. M. Hass, their representative; KOA Didak; and our Michael Taylor, who made all the arrangements with Mrs. Hass to speak to us last month, followed by the very informative tour.

It appears I pressed a panic button or two in recent statements on the membership drop. It also appears to have worked! Remember, quite a few of our hard workers, plus a few others have found it necessary to switch to other computers, or have left the area, and the work load of the club has fallen to a precious few. We need help. We do realistically expect a drop in membership, but the statement seems to have revived some of our forgetful members and the club now has between 75 and 80 members. (I hope you remembered to put up the poster in your area!) But we still require people to keep the club running at the high level it has.... There are a lot of small Canadian clubs that depend on us for news and views. Ruth still needs articles! This is where you smaller groups could come in and help us -- can you give us updates on what your club is doing and reviews of software you are using? A small sentence, a paragraph, or pages are ALL acceptable. And you local members can get more involved by helping out. The FEST is coming up fast, and the volunteers that have indicated they would help can be counted on one hand. It is impossible to put on a show that large with that little help. Donate some time; pay back the group for what it has done for you. Call now!

We would like to ask that console only users stand up and be counted. Jack McAllister is putting on a seminar this month that is just for you (details elsewhere in the newsletter), and we would like to know whether we are answering your needs. Please try and attend this seminar and talk over your requirements with Jack. He has put a lot of effort into this day and we would love to have a good turn-out for him!

We are looking for an advertising chairperson. This job entails asking the local computer and possibly other stores, if they wish to advertise in our Newsletter, arrange pick up of the ad, and collect the fee. If you would like to offer your services, contact an executive officer.

We haven't collected for a fairware author for a couple of months now. I believe that Funnelweb is widely used within our group, and as they are now distributing under Fairware it is time to acknowledge what a great job they have done over the past few years. Would someone like to collect the donations at the monthly meeting?

It also is time to take inventory of your computer and components. Been wanting that great new hardware that is being offered? Sell the other pieces rarely used at the FEST! Someone will love the "deal" you will sell it for, and you will have some spare cash to purchase new software or equipment. You might be lucky enough to find what you want on the used equipment booth....

For the March meeting, we are putting together a panel of experts to answer your questions regarding our computer. Please don't be shy. Ask! I bet there are others who have been wanting to know the answer and either forget or are too shy to ask. At one time or another, the "experts" weren't. They too had to learn one way or another. This month is for you! We are also planning to show you a piece of software... (At time of writing, there are two possibilities - a new word processor written by one of our members - no, I don't mean PRESS - or a "desk-top publishing" utility.)

Until next month...



## INTRODUCTION TO COPYRIGHT by Marcelle Gibson

Copyright is a statute which gives to creators, such as writers, artists, composers, and programmers, the legal right to be paid for the use of their creations and to control how, when and where they are used. Copyright exists for the protection of artistic invention -- whatever you create, once it is put on paper, canvas, diskette, stone, etc., is owned by you and you own the copyright on it. The creator has rights under Copyright which are bought and sold in the market place.

As you may know, Royal assent was given to Canada's new Copyright Act on 8 June 1988. This new Act was required as there had been no revision of the law since the original Act of 1924 and many inequities had arisen, mainly due to changing technology, semantics and economics: two-dimensional drawings which were made into three-dimensional objects, the negative of a photograph as compared to a print of the photograph, songwriters receiving royalties according to a 1920s compensation scale, etc. It is now considered that even though a Copyrighted item is translated into another medium, i.e., drawing to object, the Copyright still covers that item.

Further, specific mention was made of computer programs and certain sections were added solely considering matters pertaining to computers. As in the past, when a copy is bought of a programme, the right which we buy is only the use of a designated copy of the programme and not the programme itself. The new Copyright legislation now provides the buyer's legal right to make one back-up copy, which must be destroyed when the programme ceases to be their property. Please remember it is a federal offence to abuse Copyright and the fines are considerable.

A formal description of a computer program was given in the new Act in order to assist establishing legal rights. The exact wording of the Act describes a computer program as "a set of instructions or statements, expressed, fixed, embodied or stored in any manner, that is to be used directly or indirectly in a computer in order to bring about a specific result." Regarding the question of look-alikes, the source code is generally considered to be the determining factor for Copyright.

If you wish to receive compensation for your program and/or wish to protect your rights to it, you have several options to claim your rights:

In Canada, as in the United States, it is not mandatory or even necessary to register a copyright claim since copyright exists automatically as soon as a work is created. Therefore, you do not lose your copyright if you do nothing at all.

However, legal recourse is not available unless proof is established of your claim, even though your rights will exist automatically in spite of failure to register. Since all works are automatically copyrighted upon creation, when you register at a Copyright Office you are formally signifying that you are claiming your ownership of the copyright.

If you do not wish to go through the formalities and expense of registering at the Copyright Office, you may send a copy of the programme to yourself by Registered Mail. (Note: once this is received, do not open the envelope. This may then be used in a court of law to prove the date from which you have owned the right in the programme.) Be sure to keep an original hard copy of your program which is signed by you, dated, and witnessed.

You may also wish to signify your claim by clearly indicating on your program a copyright notice: the copyright symbol (the circled c), the name of the copyright owner, and the year of creation. If you do wish to register your claim, addresses and telephone numbers for the Copyright Offices are given at the end of this article.

Regarding your copyright protection in the U.S., Canadians should be aware that our laws have no application in the U.S., and vice versa. Nevertheless, there are certain mutual guarantees to both Canadian and U.S. citizens' rights in Copyright. The U.S. has had a Copyright Law since 1909, and it was most recently revised effective 1 Jan 1978. Their new Law provides protections

which are similar to ours. In both countries Copyright is protected for the lifetime of the author plus 50 years.

If you have a program which you feel should receive exceptional protection in the States, register your program in the U.S., and contact a competent lawyer, i.e., one who deals often in American Copyright matters. There are many lawyers in Canada who specialise in dealing with the arts community's Copyright problems.

For most of us it is sufficient to know that a Canadian's original work is protected in the U.S. and, in fact, almost everywhere in the world, automatically without the need for registration thanks to the Copyright Conventions.

#### The Copyright Conventions:

There are two Conventions respecting Copyright, the Universal Copyright Convention (UCC) and the Berne Convention. Virtually every major country in the world, with the exception of The People's Republic of China, is a signatory to one or both of these Conventions. Canada is a signatory to the UCC and Berne. The USA was only a signatory to the UCC until March of this year, when they will join the Berne Convention as well.

There is no registration procedure under these Conventions, nor is any formal act required to obtain protection. The only major difference between these two is that Berne does not require a copyright notice be shown on the work in order to be recognised.

Both of these Conventions guarantee that citizens of all co-signatory countries are given the Copyright protection that is applicable to other co-signatories, that is, Canada guarantees that the copyright claims of citizens of the United Kingdom, France, Japan, etc., etc., are protected under the Canadian law (The Copyright Act) which is applicable to Canadian citizens.

The information above was provided by helpful persons from the Copyright unit of the Department of Communications, the Copyright offices for Canada and the United States, CAPAC, The Canadian Museums Corporation, various artists, and my own files, and is not to be considered as an alternative to proper legal advice. If you wish further information straight from the horse's mouth, here are the addresses and phone numbers for the Canadian and U.S. Copyright offices:

#### Canada:

Copyright Office  
Consumer and Corporate Affairs Canada  
Place du Portage Tower I  
50 Victoria Street  
Hull, Quebec  
K1A 0C9

Telephone: (613) 997-1725 for personal service  
(613) 997-1726 for answering machine (to order information kits and forms)

Fee is \$35 payable to the Receiver General of Canada

#### The United States:

Copyright Office  
Library of Congress  
Washington, D.C.  
U.S.A. 20559

Telephone: (202) 479-0700 for personal service  
(202) 707-9100 for answering machine (to order application forms if you know which forms you need)

Fee is \$10 (US), payable to Registrar of Copyrights (money orders or drafts only from Canadian addresses)

## A Look At Assembler Language by R. A. Green

As I see it, programming is the ultimate computer game. It requires the intellectual activity of the toughest "adventure" games and the skill of the action games. It also requires practice as does any good game. There are several programming games that you can play. If BASIC programming could be equated to checkers, Assembler Language programming would be equivalent to chess. Although Assembler Language is more difficult than BASIC it is also more fun.

Like any game, there are certain skills and tricks that help you win. My intention in this series of articles is to show you some of these tricks and skills. I hope the articles will be of benefit to both the novice and the experienced programmer. So, dig out the rule book (the E/A Manual)! We'll be reading the rules very carefully to try to make them work for us, not against us.

The following articles will have coding examples written for use with a good macro assembler (if there wasn't one available for the 4A I'd be tempted to write one). Also, in many of the examples we will evaluate the "cost" of coding the problem. One of the goals of the Assembler Language game is to minimize the cost.

There are three costs to be considered for each piece of code. The first two are related to the machine. The first is the "storage cost" which is the number of words of memory the section of code requires. The second is the "time cost" of running the code.

Calculating the storage cost is easy -- you just count the number of words the instructions and data use. The time cost takes a bit more thought. We would like to be able to actually count the number of machine cycles needed to execute the code, but that's very tricky to do. However, on most micros, you can estimate the time cost by counting the number of words of storage that must be fetched or stored during execution of the code. Multiply and divide instructions are the exception to this rule.

Let's try an example. Consider the instructions:

```
MOV    @A,@B    Move word from A to B
MOVB   @A,@B    Move byte from A to B
```

Both of these instructions have a "storage cost" of 3 words (each instruction occupies 3 words of memory) and a time cost of 5 words. Even though the second instruction moves only one byte, we all know the 4A is a 16 bit machine and storage is always fetched or stored a word at a time even though only one of the bytes is needed. The time cost of the instructions is broken down as 3 words to fetch the instruction itself, one word to fetch A and one word to store B.

Now, a little more difficult example:

```
MOV    *R1,*R2
```

This instruction has a storage cost of 1 word and a time cost of -- quick now -- yes, you guessed it, 5 words. The time cost is broken down as:

```
1 word to fetch the instruction
1 word to fetch the address in R1
1 word to fetch the address in R2
1 word to fetch the value pointed to by R1
1 word to store the value where R2 points.
```

You have to remember, the "registers" on the 9900 microprocessor are not special (like a lot of micros); they are simply a method of addressing memory.

The third cost of code is more subjective than the first two. This third cost is the effort you have to exert to write the code and to type it into the computer. This cost can be minimized by practice and by judicious use of the tools available. One of the best tools you can use is a macro. Most of the examples in upcoming articles will use and show the use of macros.

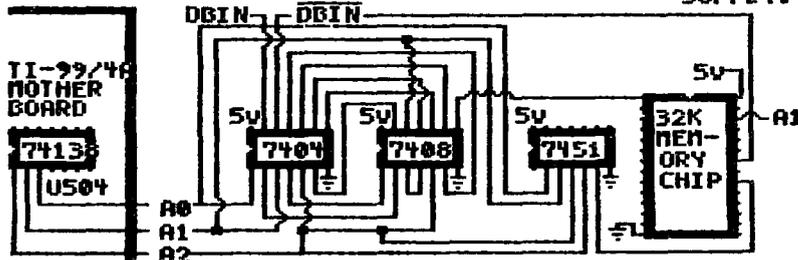
We've established the ground rules now for future articles. Next time we'll look at "switches".

"It is a mistake to allow your computer to realize that you are in a hurry, or have a deadline to meet."

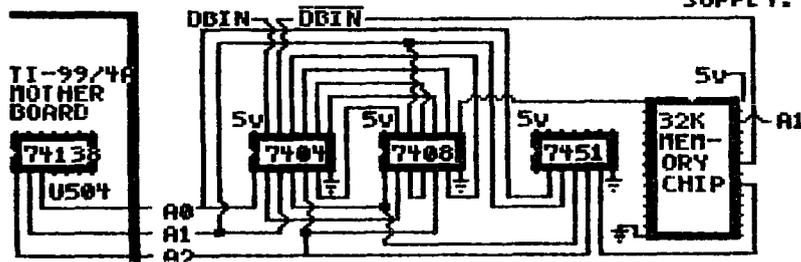
**32K EXPANSION CORRECTION FROM LAST MONTH'S ARTICLE (February 1989)**  
By David Caron

CAN YOU FIND THE DIFFERENCE BETWEEN THE PICTURES?

CHIP SCHEMATIC DIAGRAM OF CONTROL CIRCUITRY FOR 32K USING A SINGLE 32K CHIP.  
ALL CONNECTIONS BETWEEN THIS CIRCUIT AND THE TI COMPUTER CAN BE MADE USING THE GROM EXTENDER PORT EXCEPT FOR A0, A1, AND A2 WHICH ARE ACCESSED FROM THE 74LS138 CHIP CLOSEST TO THE TI EXPANSION PORT. THIS HOWEVER ONLY APPLIES TO BLACK AND SILVER CONSOLES. A0, A1 AND A2 CAN BE ACCESSED FROM PINS 3, 4 AND 5 RESPECTIVELY OFF OF THE TAL 004 CD40050 CHIP (U36) ON THE BEIGE CONSOLE.  
BY THE WAY, YOU SHOULD USE LS CHIPS (LOW POWER) BECAUSE OF THEIR LOW POWER DRAIN OF THE POWER SUPPLY.



CHIP SCHEMATIC DIAGRAM OF CONTROL CIRCUITRY FOR 32K USING A SINGLE 32K CHIP.  
ALL CONNECTIONS BETWEEN THIS CIRCUIT AND THE TI COMPUTER CAN BE MADE USING THE GROM EXTENDER PORT EXCEPT FOR A0, A1, AND A2 WHICH ARE ACCESSED FROM THE 74LS138 CHIP CLOSEST TO THE TI EXPANSION PORT. THIS HOWEVER ONLY APPLIES TO BLACK AND SILVER CONSOLES. A0, A1 AND A2 CAN BE ACCESSED FROM PINS 3, 4 AND 5 RESPECTIVELY OFF OF THE TAL 004 CD40050 CHIP (U36) ON THE BEIGE CONSOLE.  
BY THE WAY, YOU SHOULD USE LS CHIPS (LOW POWER) BECAUSE OF THEIR LOW POWER DRAIN OF THE POWER SUPPLY.



The first schematic is identical to the one in the February 1989 article on page 7. The second schematic is the corrected one.

# FAST EXTENDED BASIC

LUCIE DORRIS

Two months ago, the amount of DATA in the SHIP program may have frightened you. My program for next month (for children to play, for parents to type) also has a lot, so I want to share my PROGRAM-WRITER with you. It has been on our own TEXLINK BBS for quite a while, and was very useful for typing repetitive CALL and DATA statements. But it handled only one statement per line: handy for TI BASIC, not for XB... This new version handles multiple statements (of CALLs), variable names with or without arithmetic operators, sprite numbers, array values, and you can use it for any CALL you wish.

When you save a program under the MERGE format, Tex creates a DIS/VAR 163 file that it can read back and transform into a program in memory. To save space, Tex "tokenizes" its most used words as numbers from 129 to 254; some carry a complete word (like OPTION, WARNING, etc), others only one char. operator. Look at line 130, which defines the tokens most often used in my program as string variables: Q\$ and UQ\$ tell Tex if the following string should be Quoted or UnQuoted, i.e. shown on screen with quotations marks, or not. Tex uses the other tokens to differentiate them from ASCII characters used in text: CM\$ for the comma, LP\$ and RP\$ for left and right parentheses, QU\$ (different from UQ\$!) is the quotation mark, and SEP\$, a unique token for the double colon separator of multiple statements.

There are more tokens: DATA and CALL in lines 450 and 590, tokenized as 147 and 157 respectively. Since the arithmetic operators follow each other in the token list, we can use one POSITION statement for them all, in lines 550-570 (tokens 193-196); the last symbol in my list, # for sprite numbers, is tokenized as 253, therefore the relational expression in line 570. "ALL", as in CALL DELSPRITE(ALL), is token 236 (a freebie, line 520). A complete list would be too long: the text file TOKENS can be found on some BBSS, including ours; an early 99er Magazine published a good article on the subject, reprinted in The Best of 99'er; its example program, to do CALL SOUNDS, inspired this one.

As experienced XB practitioners, you know that CALL statements can be inherent to the Basic version you use, or you can create your own. How does Tex handle them all? Simple: it does not; there were not enough numbers left to tokenize all the CALLS graciously included in XB. A good plus, since Triton XB could add much more, and users can too; my program will take them all! When a CALL statement is saved in a MERGE format, only the CALL word is tokenized, and its subprogram-name is saved as an unquoted string. (If you ever read an XB program on disk, you will see what I mean... they are about the only comprehensible portions of the file, together with the quoted strings of your print statements....) Enough said. Here is the program:

```
100 REM ** PROGRAM-WRITER v2.0 ** L.Dorais / Ottawa U.G. / Feb. 1989
110 REM
120 DATA **ANY**, SOUND, COLOR, CHAR, HCHAR, VCHAR
130 DIM CA$(6) :: Q$=CHR$(199) :: UQ$=CHR$(200) :: CM$=CHR$(179) ::
    LP$=CHR$(183) :: RP$=CHR$(182) :: QU$=CHR$(34) :: SEP$=CHR$(130)
140 CALL CHAR(128, "0000FFFF", 129, "0000006C6C006C6C") :: CALL COLOR(13,7,1)
    :: DL$=RPT$(CHR$(128), 28)
150 GOTO 170 :: A,C,CMD,IN,L,LN,MS,P,ST,X,AS,BS,CLS,DEV$,L$,LL$,OLS,P$,PL$
160 SEP$,T$ :: CALL HCHAR :: CALL CLEAR :: CALL SOUND :: !@P-
170 FOR X=0 TO 5 :: READ CA$(X) :: NEXT X
180 DISPLAY AT(6,8)ERASE ALL:"PROGRAM-WRITER" :: CALL HCHAR(7,10,128,14)
190 DISPLAY AT(10,1):"MERGE FILENAME:" :: " DSK1.MERGE"
200 ACCEPT AT(12,7)SIZE(-12):DEV$ :: OPEN #1:"DSK"&DEV$,VARIABLE 163
210 DISPLAY AT(17,1):"FIRST LINE #?": " INCREMENT? 10"
220 CALL AC(17,15,5,LN) :: CALL AC(19,15,-2,IN) :: CALL CLEAR
230 PRINT :: DISPLAY AT(1,1):"CALL [A]ny [S]OUND [K]OLOR":
    "[C]HAR [H]CHAR [V]CHAR": "Mult.Stat.: end with a "";""
240 DISPLAY AT(4,1):"[D]ATA [R]edo [L]ine# [Q]uit": "end )" option.,
    expt quDATA":DL$ :: IF MS THEN A$=CHR$(129) ELSE A$=" "
250 PRINT STR$(LN)&A$; :: LINPUT PL$:L$ :: IF L$="" THEN 250 ELSE
    L=LEN(L$) :: CALL FIN(L$, " ,L) :: IF CMD=1 AND L>1 THEN 310
```

```

260 IF (ASC(L$)>47 AND ASC(L$)<58) OR L>1 THEN P$="" :: ST=1 ::
    IF CL$="D" THEN 410 ELSE 510
270 CALL ER :: CMD=POS("ASKCHVDRLQ",L$,1):: IF CMD=0 THEN 230 ELSE ON CMD
    GOTO 280,290,290,280,290,290,300,320,360,670
280 IF CMD=1 THEN PRINT "* Enter any CALL sub-name:" ELSE PRINT "* CHARdef
    must be >=4 chars:" or be preceded by a ":"
290 CL$=CA$(CMD-1) :: PL$="CALL "&CL$&"(" :: GOTO 230
300 CL$="D" :: PL$="DATA " :: GOTO 230
310 CALL ANY(L$,MS,X,UQ$) :: IF X=99 THEN 610 ELSE CMD=7 :: CA$(6)=L$ ::
    GOTO 290
320 PRINT "Redo last statement only:" :: P,X,C=0
330 P=POS(OL$,SEP$,P+1) :: IF P=0 OR P=LEN(OL$) THEN 340 ELSE X=P ::
    C=C+1 :: GOTO 330
340 IF LL$="" THEN LN=LN-IN
350 IF C=0 THEN MS=0 :: LL$="" :: GOTO 230 ELSE MS=1 :: LL$=SEG$(OL$,1,X)
    :: GOTO 230
360 DISPLAY AT(23,1):"* New Line:";LN;TAB(20);"Incr:";IN ::
    CALL AC(23,13,-5,LN) :: CALL AC(23,26,-2,IN) :: CALL ER :: GOTO 230
370 CALL ST(ST,L$, " ",L) :: C=POS(L$, " ",ST) :: IF CL$="D" AND
    SEG$(L$,ST,1)=QU$ THEN C=POS(L$, " ",POS(L$,QU$,ST+1))
380 IF C=0 THEN L=LEN(L$)-ST+1 ELSE L=C-ST
390 T$=SEG$(L$,ST,L) :: CALL FIN(T$, " ",L) :: RETURN
400 ! ==== data ====
410 IF L>120 THEN 460
420 GOSUB 370 :: IF SEG$(T$,1,1)<>QU$ THEN A$=UQ$ :: GOTO 440
430 A$=Q$ :: L=L-2 :: T$=SEG$(T$,2,L)
440 CALL S(T$,A$) :: P$=P$&T$ :: IF C THEN P$=P$&CM$ :: ST=C+1 :: GOTO 420
450 L$=CHR$(147)&P$ :: GOTO 620
460 CALL SOUND(150,440,0) :: PRINT : "* Your line is >120 char." ::
    L$=SEG$(L$,1,120) :: IF POS(L$, " ",1) THEN A$=" " ELSE A$=" "
470 FOR X=120 TO 1 STEP -1 :: IF SEG$(L$,X,1)=A$ THEN 490
480 NEXT X
490 L=X-1 :: L$=SEG$(L$,1,L) :: PRINT "* I will rewrite it as:":":":
    STR$(LN)&" "&PL$&L$ :: PRINT :: GOTO 420
500 ! ==== calls ====
510 IF SEG$(L$,L,1)<>";" THEN MS=0 ELSE MS=1 :: CALL FIN(L$,"",L)
520 CALL FIN(L$, " ",L) :: CALL FIN(L$,"",L) :: IF L$="ALL" THEN
    T$=CHR$(236) :: C=0 :: GOTO 580
530 GOSUB 370 :: P=POS(T$,"(",1) :: IF P THEN CALL AR(T$,P,LP$,RP$) ::
    GOTO 580
540 IF (CL$="CHAR" AND L>3 AND POS(T$,"$",1)=0) OR SEG$(T$,1,1)=QU$
    THEN GOSUB 640 :: GOTO 580 ELSE X=0
550 X=X+1 :: P=POS("+-*/^#",SEG$(T$,X,1),1) :: IF P=0 THEN IF X=LEN(T$) THEN
    CALL CD(T$):: GOTO 580 ELSE 550
560 A$=SEG$(T$,1,X-1) :: CALL CD(A$) :: B$=SEG$(T$,X+1,100) :: CALL CD(B$)
570 T$=A$&CHR$(192+P-55*(P=6))&B$
580 P$=P$&T$ :: IF C THEN P$=P$&CM$ :: ST=C+1 :: GOTO 530
590 L$=CL$ :: CALL S(L$,UQ$) :: L$=CHR$(157)&L$&LP$&P$&RP$
600 ! ==== print line to file or keep in memory ====
610 IF MS THEN OL$,LL$=LL$&L$&SEP$ :: GOTO 230
620 B$=CHR$(INT(LN/256))&CHR$(LN-256*INT(LN/256)) :: IF LL$<>" THEN L$=LL$&L$
630 PRINT #1:B$&L$&CHR$(0) :: LN=LN+IN :: OL$=L$ :: LL$="" :: GOTO 230
640 IF POS(T$,QU$,1)=0 THEN 660 ! ==== quoted ====
650 CALL ST(1,T$,QU$,L) :: CALL FIN(T$,QU$,L)
660 CALL S(T$,Q$) :: RETURN
670 PRINT #1:CHR$(255)&CHR$(255) :: CLOSE #1 ! ==== end ====
680 DISPLAY AT(10,1)ERASE ALL:"To get your program, type:" : :
    "NEW & MERGE ""DSK"&DEV$&"" :: END
690 !@P+
700 SUB ER :: DISPLAY AT(23,1):"" :: SUBEND ! erase 1.23
710 SUB S(A$,C$) :: A$=C$&CHR$(LEN(A$))&A$ :: SUBEND ! Quote/UnQuote strg
720 SUB CD(A$) :: IF A$<>" THEN X=ASC(A$) :: IF X>47 AND X<58 THEN
    CALL S(A$,CHR$(200))
730 SUBEND ! check for digit, UnQuote them
740 SUB ST(S,A$,C$,L)
750 IF SEG$(A$,S,1)=C$ THEN L=L-1 :: A$=SEG$(A$,1,S-1)&SEG$(A$,S+1,L)
    :: GOTO 750
760 SUBEND ! delete starting char. C$
770 SUB FIN(A$,C$,L) :: IF L=0 THEN SUBEXIT
780 IF SEG$(A$,X,1)=C$ THEN L=L-1 :: A$=SEG$(A$,1,X-1) :: GOTO 780
790 SUBEND ! delete ending char. C$
800 SUB AC(R,C,S,V):: ACCEPT AT(R,C)SIZE(S)VALIDATE(DIGIT)BEEP:V :: SUBEND

```

```

810 SUB ANY(A$,M,X,C$):: T=M :: P=POS(A$,";",1) :: IF P THEN M=1 ::
    A$=SEGS(A$,1,P-1) ELSE M=0
820 IF POS("CLEAR/INIT/CHARSET",A$,1)<>0 THEN CALL S(A$,C$) ::
    A$=CHR$(157)&A$ :: X=99 ELSE M=T :: X=0
830 SUBEND ! check any sub-name, process if no variable-list
840 SUB AR(A$,P,L$,R$) :: IF SEGS(A$,LEN(A$),1)<>")" THEN A$=A$&")"
850 X$=SEGS(A$,P+1,LEN(A$)-P-1) :: CALL CD(X$)
860 A$=SEGS(A$,1,P-1)&L$&X$&R$ :: SUBEND ! deal with array values

```

When you run it, you are asked for the name of your MERGE file, then the first line number and the increment. The main program starts on line 230 with a permanent "pull-down menu" of keys you can press any time the cursor is nagging at you. DATA is accessed by typing "D", and the most often repeated CALLS are predefined, just type the letter in brackets to change the input line to the call you need; remember "K" is for COLOR, since "C" is for CHAR. If this long list of five does not satisfy your needs, type "A", and Tex will show a "CALL \*\*ANY\*\*(" input. Enter the name of the CALL you want (check the spelling!), like KEY, SPRITE or MYSUB, and the next input will carry that sub-name (the line number will not change of course); you can now enter your statement, or change your mind by typing another menu command. If your ANY is CLEAR, INIT or CHARSET, which don't need a variable list, the statement is immediately processed, and the next input will still be \*\*ANY\*\*.

If you make a mistake, just enter "R" to redo the last statement, even if your current or last line contains multiple statements (I got grey hair on that one...). By pressing "L", you can change the Line number and/or the Increment, the current ones being the defaults (no room for an [I] in the menu...). [Q] is self-explanatory.

Since I wrote that program to save myself some typing, you don't have to type the ending quotations mark or ")", even in array values: "A(3),B(4))" can be entered as "A(3,B(4"; but the ending ")" is mandatory for quoted DATA statements, so the program knows how to find the next separator comma (after the ending ", not in the middle of the quoted sentence...). The menu will remind you.

If you want your statement to be part of an MS (MULTIPLE STATEMENT), follow it with a ";", easier on fingers than a ":". For example, the entry "50,440,0;" in a "CALL SOUND(" input, will instruct Tex to process it, add a separator token, but keep in memory until you tell him to print to disk by entering a statement not ending with a ";" (remember, you can always [R]edo the last statement). In the case of CLEAR, INIT and CHARSET, the ";" must be added to the word itself, as in "CLEAR;". Just be careful not to switch from CALL to DATA in an MS line, or change the line number! Unpredicted results... You will be reminded that you are into MS mode by a red double colon after the line number in the input line. The program does not check the length of multiple statement lines, only the DATA ones, so be careful.

Three last notes: since single letters are used for menu commands, you cannot enter single variables: CALL SCREEN(C) will bring a CALL CHAR input, and a single letter which is not a menu command is not accepted; just use a numeric value, or a two-char. variable name, then change it when you complete your program. CALL CHAR definitions must have at least four characters (otherwise Tex would read "123" as a numeric value; use trailing 0s if necessary), or be preceded by a " (as in "0 or "FF). If you build your MERGEable file in sections, don't forget to change the subsequent filenames. Otherwise, the new MERGE (if you keep my default) will overwrite the first.

I wondered what happened if I typed numerous lines with the same line number: the MERGE file, as read by a disk editor, contains them all (the same thing happens if you [R]edo a one-statement line), but Tex will keep only the last one as the valid one for the program. So you can always redo a previously saved line (yes, all those multiple statements...), but you must use the [L]ine# function, not the [R]edo one, which applies only to the last statement entered. You can also type the lines in any order you wish. Tex will put them in order when it merges the file.

Comprehension of the listing should go smoothly to line 250, which LINPUTs (the commas!) the one letter command or our data or values, and sends the ANY sub-name inputs to line 310, where CALL ANY looks for CLEAR/INIT/CHARSET; if found, flag X is set to 99; if not, MS is restored to its temporary value T before leaving the sub, and CMD becomes 7 to be used by line 290.

Line 260 divides the flow between DATA and CALL, while line 270 deals with single letter menu commands. Lines 280 to 360 offer almost no difficulty; CL\$ contains our CALL sub-name (CHAR, SPRITE, and the like): the six basic ones, or the one entered by you, temporarily stored in CA\$(6). The only typing problem you might encounter will be the quotation mark; to be included in a string (more quotation marks), it must be doubled: watch for them in lines 230, 240, 280 and 680. DATA statements are first checked for length; if found guilty, they will be cut at the last comma or space before the 120th character (lines 460-490).

The sub in lines 370-390 works for both CALL and DATA; it cleans up starting spaces (CALL SStart), looks for the next comma in your statement, or the end of it (if a quoted DATA statement, it will check again after the ending quotation mark), then defines T\$ as the segment to be worked on and cleans up ending spaces if any with CALL FINis. Back in the DATA portion of the program, if your statement is not quoted, A\$ becomes CHR\$(200); if quoted, A\$ becomes CHR\$(199), and length L and T\$ are changed to get rid of the ""s (Tex will give them back to you when it encounters CHR\$(199) before the string). In line 440, we CALL S, a defined sub that rebuilds the string T\$ as Tex likes it: the Quote/Unquote character, the length of the string, then the string itself. T\$ is then joined to the previous portion P\$ of the statement (before the comma); if there was a comma after the statement (C>0 in the sub 370), it adds the comma token, CM\$. The starting position ST is then incremented, and we go back to line 420 until the whole statement has been processed. When all is done, line 450 adds the DATA token at the beginning, and sends the resulting L\$ off on its destiny in line 620.

The CALL statements are handled almost the same way, but here we deal mostly with numeric values, variable names or array values (always preceded by a variable name). Tex handles digit values as unquoted strings, while variable names, like A or A\$, are left alone (did you ever wonder why variable names must start with a letter?). CALL KEY(0,K,S) will save "0" as CHR\$(200)&CHR\$(1)&CHR\$(48), but "K" is saved only as CHR\$(75). But before we check for digits with the CD sub, we first look for the multiple statement symbol ";", and set the flag MS accordingly. The two CALL FIN will delete the ending )s, and extra commas (if you forgot your last numeric value, Tex will warn you only when you run the program...).

As in DATA, the sub in line 370 will get a T\$ segment to deal with. If it contains a (, it is an array: SUB AR will give you back the trailing ")" taken off by SUB FIN (just to adjust the length for next line), and will rebuild the statement with the proper tokens for the parentheses; please don't include any arithmetic symbol in such a line, it will not be processed properly (deadline was two days ago, and my head is hurting): CALL SCREEN(SC(X)) and CALL COLOR(S(3),F(4),B(X)) will work, but not CALL SCREEN(SC(X)+V-W).

If the segment is a CHARdef (four char. or more), or quoted by you (only starting " is needed: CALL SAY("AMORE is OK), the sub 630 will delete the ""s if you typed some, and transform the T\$ string into a Quoted one. If not quoted, line 550 takes the whole T\$ segment and looks for a symbol; this takes some time: each character must be read until a symbol is found, and it takes longer if it finds none... which means that we have a numeric value (checked by sub CD and "UnQuoted" accordingly) or a simple variable: line 580 will add it to P\$ and commatize it if necessary.

If there is a symbol, we have things like "T/2" or "X+Y", i.e. a combination of values and variables: line 560 will cut T\$ in two, check each part for digit (and UnQuote accordingly), then will rebuild T\$ with, between the two parts, the tokenized symbol. The program cannot handle rare things like "T+V-3", i.e. more than two values/variables; please avoid them, or edit the line when you merge your work as a program in memory; the line will look OK on the screen, but will not be, since some characters will have their ASCII value, not their tokenized one. Finally, line 600 "unquotes" the CALL sub-name itself, renamed L\$, with a CALL S (CL\$ is left untouched for the next input). L\$ is then rebuilt with the CALL token, its unquoted sub-name, a left token parenthesis, the processed statement P\$, and a right token parenthesis.

The statement is now ready; if we are in "MS" mode, it gets a separator SEP\$, and is kept in LL\$ (our current MS line) and OL\$ (our last line, or current until printed to disk) for use by [R]edo. Otherwise, save the line to disk: line 620 tokenizes the line number LN by converting it to the characters of the decimal equivalents of two hex bytes, then it builds the MS line if LL\$ is not

empty, and the whole line is finally sent to the MERGE file with the end of record marker "CHR\$(0)". Line number is incremented, OL\$ now holds our complete printed line, and MS line LL\$ is emptied.

Ouf! Time to [Q]uit: when you do, Tex will properly close the MERGE file with an "End of file" marker, CHR\$(255)&CHR\$(255), i.e. >FF. If you break the program and type CLOSE #1 in direct mode, the file will merge OK, but you will get an I/O ERROR 25 message, "Input File past end"; enter line 680 in direct command mode to close it properly.

### News from Other Groups

A letter dated December 3, 1988 was passed to me by Dave Morrison at the January executive meeting. I intended to share it with you in the February newsletter, but it didn't fit in conveniently. Here is part of the letter, the part in which Jan is seeking to share information on a project. I hope you will find the letter to be of interest, and encourage you to participate by replying if you have any knowledge to share. Projects like this show us that there are still areas to explore in the TI world. Whether or not the project is successful, it will perhaps inspire others to conceive similar group projects.

Dear Ottawa Users Group:

As a member of the Chicago Users Group, I am happy to correspond with others who take their TI's far above the heights that TI left us with. At times, I wonder if we are not all better off since TI left us orphans, since necessity has given us the drive to create products. Ramdisks are just one example of the third party support that gave us speed and flexibility.

I have installed a 256K ramdisk (this particular ramdisk can be upgraded to 512K) inside a black TI console (using ROS 7.3). I am not the first to have done this. I ran into a problem when I tried to separate the console/32K/Ramdisk from the Expansion box. It seems that the Ramdisk ROS is made to co-exist with the DSR of the disk controller, and without a disk controller, the execution of programs in load and save will crash the system. In talking and writing with others, I have found that others have installed Ramdisks in consoles, in an attempt to make a PORTABLE, and have all run into this. I have been informed that a Mr. Pat Saturn in Ohio has overcome this in conjunction with Ron Gries. Today I was informed by Mr. Bud Mills that his company is in the process of writing a program to help me. So whereas I am temporarily stumped with the problem of running a ramdisk without a disk controller, this problem looks to be a temporary one.

I would like to correspond with anyone who has attempted to install a ramdisk inside a console, with the intention of sharing information on our successes and failures. I would think that this would improve our experimenting efficiency. I write this on my own, but will submit all information received to your User Group, as your information will be submitted to the Chicago Users Group. This Special Interest Group focusing on PORTABLE operation of the TI will cover such items as converting a TI to battery power, 32K memory installation, ETC. If any of your members wish to share information, please pass along my mailing address.

Jan Janowski  
8536 N. Keystone  
Skokie, Ill 60076

TI BASIC continued from February  
by Steven Shaw

Many printers now permit you to "download" a screen display, using '8 pin bit image mode'. Most of the EPSON range of printers permit this for instance. Programs to USE these facilities are rare however, and one is therefore given below.

This is in TI BASIC, but requires the MINI MEMORY MODULE to function. The program will also run in EXTENDED BASIC.

Because the printer defines its characters vertically, but the computer defines its characters horizontally, this program will produce a picture of the screen on its edge : it is easier to do this than to rotate the image.

If the screen contains only text there is no advantage to using this program: Use GCHAR to obtain the characters and PRINT them one row at a time.

This program will print the characters as they appear on screen:

- i. IN BLACK ONLY. ON pixels are printed, off pixels are not.
- ii. Sprites are not copied.

```
100 OPEN #1:"PIO.CRLF"
110 REM OR EQUIVALENT RS232 FILE NAME
120 PRINT #1:CHR$(27);"A";CHR$(8)
130 FOR [A=1 TO 32
140 PRINT #1:CHR$(27);"K":CHR$(192);CHR$(0)
150 FOR [B=24 TO 1 STEP -1
160 CALL GCHAR([B],[A],[CHAR)
170 IF [CHAR<33 THEN 300
180 CALL CHARPAT([CHAR,DEF$)
190 IF DEF$="0000000000000000" THEN 300
200 FOR [SEG=16 TO 2 STEP -2
210 [HEX=ASC(SEG$(DEF$,[SEG,1))
220 GOSUB 430
230 [PRINTDEF=[HEX
240 [HEX=ASC(SEG$(DEF$,[SEG-1,1))
250 GOSUB 430
260 [PRINTDEF=[PRINTDEF+[HEX*16
270 PRINT #1:CHR$([PRINTDEF);
280 NEXT [SEG
290 GOTO 340
300 FOR []=1 TO 7
310 PRINT #1:CHR$(0);
320 NEXT [
330 PRINT #1:CHR$(0)
340 NEXT [B
350 PRINT #1:CHR$(13);CHR$(10)
360 NEXT [A
370 PRINT #1:CHR$(27);"@"
380 PRINT #1:CHR$(7)
390 CLOSE #1
400 REM NOW GO WHERE YOU WISH
410 REM USING RETURN OR GO TO AS APPROPRIATE
420 STOP
430 [HEX=[HEX-48+([HEX>64)*7
440 RETURN
450 END
```

To permit the routine to be added to any program, the square bracket has been used in front of each variable: [. The [ is accepted as a valid character in variable names.

ESC K {CHR\$(27);"K"} is used in the Epson printer, and some others, to select: 'Normal density 8 pin bit image mode'

## HOTLINE NUMBERS

The executive has expressed a desire to assist all members should you have some problems or questions, want to do some library swapping or borrow a book. This will be the place to look. Listed here are the members of the executive, committee heads, and others in the group willing to help in their specialized areas. Of course, if you wish to be placed on the list, just give me a call. I know there is a lot of expertise within our Group, so I hope to add to this list. Please respect normal hours unless you specifically know that someone doesn't mind a call at 3am, or use the BBS to leave a message at 738-0617, 24 hours a day, 7 days a week.

JANE LAFLAMME.....PRESIDENT.....(H) 837-1719 or (W) 745-2225  
 AL PALMER.....VICE PRESIDENT.....594-9216  
 MARCELLE GIBSON.....SECRETARY.....233-2384  
 BILL SPONCHIA.....TREASURER.....523-0878  
 MICHAEL TAYLOR.....PAST PRESIDENT.....831-0143  
 PETER ARPIN.....SYSOP.....523-0017  
 RUTH O'NEILL.....NEWSLETTER EDITOR.....234-8050  
 TONY HOPKINS.....ADVERTISING.....746-4463  
 DAVE MORRISON.....LIBRARY CHAIRMAN.....737-4889  
 JACK McALLISTER.....CASSETTE LIBRARY.....225-6989  
 HENRI MONAT.....ARCHIVES.....824-0941  
 LUCIE DORAIS.....MEMBERSHIPS.....232-0393  
 BOB BOONE.....HARDWARE/SOFTWARE.....(705) 476-9391  
 ART GREEN.....ASSEMBLY HELP.....837-1955  
 DICK PICHE.....TECH.....521-8667  
 CLUB BBS.....SET MODEM TO 8N1.....738-0617

**1989 RENEWAL**  
**\$ 25.00**

**NEW MEMBERS**  
**\$ 25.00**

NAME \_\_\_\_\_

ADDRESS \_\_\_\_\_

CITY \_\_\_\_\_ PROVINCE/STATE \_\_\_\_\_

POSTAL CODE \_\_\_\_\_ TELEPHONE (\_\_\_\_) \_\_\_\_\_

Please make cheque payable to the Ottawa TI-99/4A Users' Group and send it, along with this form, to the address shown on the cover page -- or better still, bring both to a meeting.

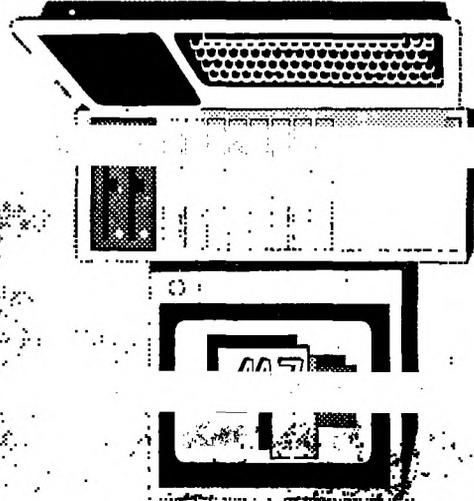
DEALER ENQUIRIES WELCOME

CANARIA DATA INC.  
264 Weber St. W.  
Kitcheners, Ontario  
N2H 4A6  
(519) 578-3873

COMPUTER DOWNLOAD  
UNLIMITED  
8 Talon St.  
North Bay, Ontario  
P1A 1N5  
(705) 476-9391  
Data (TEXLINK) 476-3043  
Compuserve 73657,3617  
DELPHI I.D. "CDU"

DATA\*PORT  
2846 Göttingen St.  
Halifax, N.S.  
B3K 3E1  
(902) 454-0232  
Data (TEXLINK) 455-2076

Support your Canadian dealers:



Authorized Distributor for the following:

LAFAMME & WRIGLEY  
Ottawa BBS (TEXLINK) (613) 738-0617  
DELPHI "JANLAFAMME" (No space)  
Compuserve 76046,2006  
DATA (Phones):

FROM

P.O. BOX 2144, STATION D, OTTAWA  
\*\*\* ONTARIO, CANADA K1P 5W3 \*\*\*

