(065)
8809

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

NEWSLETTER

VOLUME 7 NUMBER 07.....SEPTEMBER 1988

DON'T FORGET THE MEETING -- September 6, 1988

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

COMING EVENTS

September Meeting:	September 6, 1988 7:30 p.m.	Merivale High School Come on out for the first meeting after the holidays!
Multiplan Workshop:	September 17, 1988 10:00 a.m.	Merivale High School Contact Bill Sponchia for further information.
October Meeting:	October 4, 1988	Merivale High School Election time again! Don't forget to contact Bill Sponchia if you're interested in running!
Newsletter Deadline:	September 15, 1988	(or any time before that!)

EDITOR'S NOTES
from
Ruth O'Neill

It's hard to believe the summer went by so quickly. The barbecue at O'Connor's seems like just yesterday, and it's time for the September newsletter to go to the printer. The barbecue, by the way, was definitely a success, thanks to our gracious hosts. The turnout may have been a little smaller since one of the more impressive storms this summer postponed it to the holiday weekend, but it was worth the wait.

Lucie's cover art this month is a salute to some of our members in other countries, including Jiang Huixia from China. The highlight of the June meeting for me, and, I am sure many others, was meeting him. The club and individual members had a number of gifts for Jiang Huixia, including a membership to the Ottawa Users' Group.

Texlink is up and running at 2400bps, now! After a favorable review in Micropendium and a Compuserve Conference, we've had a number of orders for our favorite BBS, and hope to see more.

On the subject of telecommunications, the Ottawa group has been in the habit of logging on to Compuserve regularly for the Sunday night conferences. At one point, I asked Jim Horn (the sysop there) if he thought Compuserve minded all of us coming in under one i.d. He said that, on the contrary, he thought it was great, and that if we could get a few more groups like that into the conferences, we'd have some really interesting co's. This, then, is directed at other users' groups: If someone in your group has a Compuserve i.d., why not get a group together and attend a conference? At 300bps (which is all you really need for conferencing), with a few people sharing, the cost would be quite negligible. Maybe the club could even afford to chip in for the first one. If no one has an i.d., consider getting one for the club under a member's name. These conferences have been exciting and EXTREMELY informative. Topics are usually posted well in advance. Most formal conferences are on Sundays, but are often held on other days, too.

Delphi has also had some interesting conferences, although as yet they aren't as regular as Compuserve's. It also has a special section just for us Canadians -- Canada Communique, both a message base and file area. Jeff Guide, the sysop there, has asked me to remind Canadians of this special service, and is looking for ideas for Canadian Conference guests. If you're thinking of joining Delphi, btw, now's the time. There is a special signup offer in effect for the TI NET there -- Save $30.00 by joining now for $19.95, which includes the 500-page "Delphi -- The Official Guide" and one hour of connect time. To sign up online to Delphi and TI NET -- Dial 800-365-4636 or 617-576-2981. Once connected, press return twice. At PASSWORD prompt, type TISPECIAL and press return. If you're interested in further details, I have the full press release. Give me a call at 234-8050 or contact: Jeff Guide, System Manager, TI NET, P.O. Box 244, Lorton, VA 22079.

The President's Two Cents' Worth
by Michael Taylor

Now, suddenly, September is upon us and my plans for the summer recess are far from fulfilled. That thief of time, procrastination, is my undoing -- but I blame the stifling, muggy weather for my lethargy. Fortunately, not all of your executive was as moribund as I.

The executive meeting in July was held at Moosehead Lodge. Again, as in past visits, our hosts, Mal and Mary Taggart, received us with warmth and humour. I hope the executive will continue this "Moosehead" tradition. I hope, too, that more members will participate. The cost is modest; the atmosphere relaxed; the setting beautiful; and the food substantial (after all, it is a hunting and fishing lodge). There was a good electrical storm during our stay, but it took place mostly during our meeting on Saturday afternoon and did not cramp our style. Besides the more mundane aspects of the meeting, we were also entertained by a race between our Veep and Peep. While this race was aquatic, rather than political, perhaps we will see a rematch in the forthcoming club elections?

It is not too soon to give some thought to the club elections. Are you interested in acting in an executive capacity? If so, let the chairman of the nominating committee (that's Bill Sponchia) know. Bill will try to call every member to ask if they wish to stand for election, but if he misses contacting you, don't be shy -- give him a call.

This summer also saw the departure of Bob Boone from our immediate sphere. Our loss, and a real loss it is, is North Bay's gain. Once more, Bob, thank you for all you've done -- for the Ottawa club, and for the TI community. We look forward to those visits you promised us (and we hope you can cram some Computer Download goodies into your auto trunk for us when you do come).

Our Past President, Bill Sponchia, is organizing a series of Saturday morning workshops, to be held monthly at Merivale High School. The subject of these workshops will be announced in advance. Bill has some ideas for topics, but if there is something that you particularly would like to work on, or if you have some expertise you would be willing to share, please talk to Bill. The first workshop will be Sept. 17, at 10 a.m. Bill, in concert with Jane Laflamme, will describe, demonstrate, and teach the use of the spreadsheet, Multiplan. This will be a "hands-on" tutorial, suitable for the experienced user as well as the beginner. If you want to learn how to use Multiplan, or how to use it more effectively, please plan to attend.

The Central PA 99/4A Users Group has written to announce their 1988 Computer Exposition, to be held in partnership with the Cumberland County Amateur Radio Society, on Sunday, October 16th, from 7am to 2pm at the Cumberland Fair Grounds, Carlisle, Pennsylvania. If you are thinking of attending, leave a note on the Bulletin Board, or let me know. There may be members who would like to join you, and perhaps to share expenses.

Besides the general meeting, the assembly language workshop, and the Saturday tutorial, there is another club get-together. This latter is less formal and less regular, but it is as informative and as entertaining as the others. I refer to our telecommunications conferences. These are usually on Sunday evening, but sometimes on week-nights. So far the venue has alternated between Jane Laflamme's home and Ruth O'Neill's home. TIers from across Canada and America get together to exchange information and ideas. It is quite an interesting thing to watch (even more interesting to actually participate). If you would like to learn what goes on at one of these teleconferences, just keep an eye on our BBS for a notice and give the host a call.

BROWSING THE LIBRARY
--with DAVE MORRISON

Although the library remained open during the Summer months, the activity closely resembled the weather -- a severe drought! No requests for disks, programmes or information were received, nor were we the recipient of any new material.

I realize that many members find that the contents of our library is disturbingly similar to their own library, or that the software in our library is not what they would like. However, as I have pointed out previously, our library is essentially a product of our members. At the present time the material in the library appeals more to our newer members and to those who have recently upgraded from cassette to disk systems.

User Group newsletters (both domestic and foreign) have indicated a much higher level of software donations to their libraries than is evident in our group.

Perhaps those members who now possess extensive software libraries would consider providing the library with a list of non-copyright material that could be copied for inclusion in our library. Additionally, those members who access other Bulletin Boards might consider offering copies to the library, of any files that they have downloaded over the past several months.

Spad XIII Mark 2 -- A Review
by Ralph Kuhn

If you've tried some of the other flight simulators for the /4a, you'll be pleasantly surprised by Spad XII Mk. 2. It has a range of new features over the Mark 1 version that make it well worth the minimal cost to upgrade.

The old instruction book still applies, and is included in the package along with the all-important addendum. This describes the new features in detail, but some improvements are not given enough credit.

For those who have never piloted a plane before, this is an enjoyable way to learn about a few of the major effects that cause a plane to fly (or crash). No need to worry, though. Simply press the 'S' (start) key and the plane is back on the runway, gassed up and rarin' to go.

The old controls are the same, but there are a few changes in the effects of a few of them. For instance, the plane will turn when the stick is pushed and then slowly it will try to level out. For those in the know, this is called "a slightly positive roll stability".

Improved graphics are mentioned. I found recovering from a stall in Mk. 1 very disorienting, because unless I caught a glimpse of the horizon line, I couldn't tell which way was up. Spad Mk. 2 shows ground markings and also offers a remote view (the I key) so you can see what's happening from outside the plane.

The 'Move' key will attract those who frowned on the 10-15 minute trip needed to get your Spad over the German lines for a dog-fight over enemy territory. It magically pops you into different areas of the countryside such as the Eiffel Tower, the German airfield, the French airfield (home!) or it can be used as a convenient 'pause' for those necessary trips the pilot has to make (meals come to mind).

Along with all these improvements (and I'm sure I've missed a few) is the addition of a Fokker triplane as an enemy. If you're unfortunate enough to discover this devil on your tail, you'd better have some slick moves up the sleeve of your flight jacket. He will hunt you down and try to make a smoking ruin of your Spad.

The disk is a Not-Polyoptics product available from Computer Download (Mr. Boone does still visit the area) or your local dealer. See if you can become an 'ACE'.

An Open Letter to Fairware Authors
by Ruth O'Neill

Dear Fairware Author:

Fairware (shareware, freeware) is a wonderful thing for the user at large. Being able to try your software out before paying for it is a special privilege for a computer user, and one that I would like to see continued. I realize that this privilege is all too often abused, and that it is quite possible that you will not receive just compensation for your work. I do believe, though, that there are certain things that you can do to help us, as users, to understand your position better, and that will increase the likelihood of our paying you for your work. These are the things that _I_ would like to see in fairware programs:

1) Clearly indicate how the program is being distributed. By all means, put a "copyright 1988 your name" in the title screen, but if the program is either public domain or fairware, tell us that, too. Too many programs that authors might like to have distributed freely have been left out of user group libraries because their status was uncertain. If you want to make your program public domain, and expect no money in return, good for you! That is a generous contribution to our community. On the other hand, if you have chosen to distribute your program as fairware, please make that clear too.

2) Put this information in the program_, perhaps in the title screen, where it is easy to find. It isn't necessary to be overly obtrusive, but it should be possible to check the status of your program by running it up and looking in an easily-accessible place. Make sure your address is there, too. This information can be put in the documentation, but unfortunately it is all too easy for the program to become separated from its documentation. Another sad fact is that the documentation is all too often left unread, especially on those programs that are sufficiently well-written that they can be run fairly easily without reading the documentation thoroughly. Our failure to read the documentation is certainly not your responsibility, but you should keep that possibility in mind if you expect us to know that you want us to pay you for your work.

3) Don't give us an easy out. Tell us to pay you if we _USE_ the program, not if we _like_ it. It's all too easy to rationalize not paying you if we have to like the program before we pay you. We can use the program every day, and still be slightly unsatisfied if it doesn't perform a certain function precisely the way we would like it to. You still deserve payment in such a case.

4) Suggest an amount to send in. Give us specific guidelines. Put a price on your work. Having to decide how much to send you will probably delay payment, as we use the program longer and longer so we can properly decide on its worth. You don't have to _insist_ on a certain amount, though. You may want to tell us that any amount is_welcome_, but please give us some guidelines.

5) Don't undervalue your work. We in the TI world are fortunate in the amount of software that is available to us at prices far below those paid by owners of other computers. Most of you will have spent a considerable amount of time and effort writing your programs, and that should be recognized. Don't ask me for less than $5.00, either. It's hard to picture my 2 or 3 or even 4 dollars making a difference to someone's life, so I suspect that most people won't place sending you a very small amount of money very high on their list of priorities. I know I'd feel much guiltier delaying a payment of $10.00 than of $2.00. Besides, if your program is worth paying for, you _deserve_ more than that.

None of the above are actual requirements for fairware. They are simply things that make it far more likely that I and, I believe, many other users will pay you for you work.

And now a note to any fairware users reading this:

Make a list of the fairware you use, and check off any that you have sent in your contributions for. For the rest, if you actually use them, whether or not you truly like them, whether or not the author has been especially clear about what should be sent, start sending out cheques!

AUTOEXEC *VERY* QUICK TUTORIAL
by Jane Laflamme

This is a very quick tutorial on how to create an AUTOEXEC file and a few ideas on how to manipulate it. The following is my autoexec file and I have put the reasons for it in brackets. Of course, if you want to copy this, don't use the text in brackets!!!!

```
ECHO OFF
MODE 80                                                   (80 column text screen)
SPOOL 16                                                 (Print Spooler installed)
RAMDISK 180                                (Double Sided, Single Density RAM disk)
ASSIGN C=DSK5:                            (You have to give it a letter to access)
ASSIGN E=DSK6:                            (           ditto                      )
ECHO DSK5 is Drive C and double sided..          (To let me know they            )
ECHO DSK6 is Drive E                             (   are installed               )
ECHO Print Spooler installed.                    (ECHO will print text to screen)
ECHO                                            (This will echo blank if no text!)
ECHO Good Morning... Have a good day!      (Makes me feel good in the morning)
ECHO
A:CONFIG-16                (This command will load pgm. into MDOS from Drive A.
                            Once the program CONFIG-16 is exited, the Autoexec
                            will take back control.)
ECHO
ECHO
ECHO Hit any key twice for GPL -
ECHO Ctl. C to abort GPL load.
PAUSE                                          (To allow for the Ctl. C command)
PAUSE                                                            (To go into GPL)
ECHO ON          (Following command will now be printed on the screen so you
                  will know that it is being executed...)
E:GPL
```

Other commands you might like to try are in the manual -- eg: The PROMPT command will change the prompt in MDOS. You can put the date, time, or use DSK(n) rather than letters. Experiment.

As in the above example, the Autoexec file can "run" another file. The following autoexec file will create a menu that will executed the file "M". First create your AUTOEXEC file:

```
MODE 80
ECHO OFF
ECHO MENU
ECHO ----
ECHO
ECHO A....CONFIG-16
ECHO B....GPL
ECHO C....MYART
ECHO
ECHO (Type "M" followed by a space, and then the letter of the
ECHO  selected program...)
```

Save the above from MY-Word with the print file (PF) option, and strip control characters with the "C" command, eg: "PF" <ENTER>, "C DSK1.AUTOEXEC" <ENTER> or... Type it directly into MDOS with the command "COPY CON AUTOEXEC". (COPY whatever follows, typed on the CONsole, and name it AUTOEXEC. When you have finished typing, press Control-Z to save it to drive A.) Now create the "M" file:

```
IF %1==A DSK1.CONFIG-16
IF %1==B DSK6.GPL
IF %1==C ECHO INSERT MYART DISK INTO DRIVE A
PAUSE
A:MYART
```

(Note: If you select option C from the AUTOEXEC file, I have paused the execution of MYART to allow you to switch disks. Also, if you use option B, you must assign DSK6 as E, as in my first example, which of course, I didn't do in the "menu" autoexec file... ooops ... but as this is a very quick tutorial.....)

Save this file, as you did with the Autoexec file, with PF option of MY-Word OR the COPY CON M command from MDOS. These are just a few ideas on how to manipulate an AUTOEXEC file. Now have fun and create your own. There are many ways and ideas in the manual. Also you can run another AUTOEXEC file from a different disk if you are STILL in MDOS, by pressing Control, Alt, Del, altogether. If you come up with a great idea, (or even not-so-great), we would really appreciate you uploading to the BBS!! Of if you have any questions, don't hesitate to ask. Someone else might be too shy and you'll help them too.

The Electronic Eye
by Charles Earl

Well folks, I thought it was time to gather up some of the more interesting programming suggestions for the Geneve from Delphi's extensive message base.

When a programmer writes for both machines it is nice to take advantage of the extra capabilities of the Geneve computer. A common method suggested for Geneve detection consists of reading in the contents of address >0000, then trying to alter the value at >0000. If the value is alterable then obviously a Geneve is in use. The programmer must remember, though, to restore the contents of >0000 to the original value.

Another, possibly better, solution would be to write a value to >8100. This is echoed to >8300 on a TI-99/4a, whereas a Geneve has ram located there. Remember to backup the original values at >8100 and at >8300 before doing this. Always restore the addresses back to their original value.

Direct keyboard access is achieved differently on the Geneve than on your TI-99/4a computer. The keyboard register is located at >8008 in the TI-99/4a mode and at >F118 in MDOS. Check the cru bit at address >0008 for a value of zero, which indicates a new key press. Copy the value from the keyboard register, then reset the keyboard with a SBZ @>1EF2 then a SBO @>1EF2.

One small comment on 80-track drives and cataloging disks -- Many programs catalog disks using the direct sector access method. This causes problems with 80-track drives, ramdisks and hard drives. For those of you who wish to use this method anyway, each bit in the bit map accounts for 2 sectors instead of one. This method means that file allocation is done in 512 byte blocks.

Don't forget the trading post!

It's here for you!

Archiver III -- A Brief History and Review
by Charles Earl

First, the History:

A few years ago, Barry Traver created a method of "packing" any type of TI-99/4a files into one large file. This allowed for more convenient transferring of files between computers. Nothing is more irritating than downloading 10 files from a bbs only to discover that the package won't run because a file is missing. Unfortunately there were two drawbacks to "packing" files. First, the pack/unpack program was an extended basic hybrid which was slow. The other problem was the lack of file compression.

Later on, Barry Boone took the concept provided in Barry Traver's original program and created a compatible assembly pack/unpack program, which provided a significant increase in speed -- but his program still lacked a file compression utility. Barry Boone then released a pack/unpack program that compressed the archive. The popularity of this was outstanding, even though it was necessary to first pack the files and then compress the archive, which was time consuming.

This August, Barry Boone released Archiver III, which packs and compresses in one step, a long-awaited feature. The ability to pack and compress in one step was not the only impressive feature in this program. The time it takes to compress files is reduce significantly, and the compression routine has been improved to compress the files even smaller than with previous versions. The catalog disk and catalog archive file has been improved to include a hardcopy option which is very useful.

And Now the Review:

Immediately upon hearing that Barry Boone had released Archiver III, I logged onto Compuserve and downloaded it. Why the excitement? Well, there are two major programs required for telecommunications: a good Terminal Emulator, and a good archiver. Archiver III is an excellent archiver.

Archiving provides, on the average, a 50% reduction in file size, and thus a 50% reduction in time to transfer a file -- when you are downloading from a long distance bbs or from a major network, your pocket book will appreciate it. Unfortunately, if the archiver is slow, you sometimes wonder if the money saved was worth the time involved in unarchiving. Well, Archiver III has removed any doubt in my mind!

Archiver III will pack and compress files in one step, unlike the older Archiver II. With Archiver II you had to pack the files then compress the archive -- a slow and time-consuming process. Not only that, but packing and compressing required a significant amount of disk space. You had to have the same amount of disk space available as the total size of the files you were packing as well as enough space for the compressed file. Now, with Archiver III you only have to have enough disk space for the compressed file. It is also possible to extract a single file from an archive.

In addition to the major improvements, Archiver III also produces smaller archives, is faster and easier to use, and you can print out the catalogs of archives to your printer. Barry improved the compression routine by replacing the "junk" commonly found at the end of each sector of certain file types with zeros. The zeros compress to next to nothing.

An additional handy feature of Archiver III is the View file option, which Allows the user to extract a d/v 80 file and read it without leaving Archiver.

Unfortunately Barry, who is currently a full time student, is becoming discouraged at the lack of response to the archiver program. Many people have said that they wouldn't send in their contribution to Barry until the program packed and compressed in one step. Well, I hope the people who said that will send in their contributions now and encourage him to not only improve this program, but to produce more programs for the TI community.

FAST
EXTENDED BASIC

LUCIE DORAIS

Back to School time... and back to Extended Basic..., but I will make the first class easy: no tutorial, only a fun program. Actually, it could help you in your real-life History classes, since it will print a CALENDAR for any year from 1583, when we changed to the present calendar. No limit on the future!

I wrote this program two years ago to help me in my real-life profession as an art historian. Just for anecdote, here is one way I used it recently: I was studying a sketchbook used by a Protestant minister... most of his sketches were dated. Question: did he "sin" by drawing on Sundays too? The years were 1844 to 1846. And yes, he did draw on Sundays too, although this was a hobby for him, his main duty being preaching the Protestant Faith to French Canadians... with not much success, as history has since proven.

The program worked fine, but the calculation of each month's calendar took ages. The routine itself, copied from a TRS-80 program by Bertrand Legault from Montreal (Computing Now!, October 1983), first fetched the name and last day of each month (28, 30 or 31) from 12 "IF M=n" statements, then it "built" a long string for each month, backwards, from the last day to the first. The printing portion was mine; for some obscure reasons, I was building three months at a time, one string for one week (a two-dimensional array of 3x6), then printing them. The whole process took 45 seconds to build those three months, then goodness knows how many more to print them, and it repeated four times.

When I finally talked myself into acquiring a printer buffer, I thought of optimizing the program, at least for the printing portion. But I also worked on the calculation portion itself, eliminating some FOR-NEXT loops (lines 240-290, 420), then defining a constant "month-string" MO$ with all the days up to 31, and four other constant strings, LM$(), for the names of the months. The value of M is enough for Tex to understand which month to work on! All that was left to do for each month was to read the MO$ string to the proper position found in the LD (Last Day) array. Miracle: the calculation went down to... only six seconds for the whole year, and the printing portion was easier to do, with 12 strings of identical length to deal with. And since I now intend to start a small "business" of printing calendars for my National Archives colleagues, I made the program bilingual.

```
100 REM ***********************
110 REM   PRINT A CALENDAR
120 REM  L. Dorais(3-86/7-88)
130 REM ***********************
140 REM
150 CALL CHAR(95,"000000FF") :: E$=CHR$(27):: L$=" "&RPT$("   "&
    RPT$(CHR$(45),22),3):: S5$="     " :: S2$=RPT$(" ",12) ::
    S8$=RPT$(" ",18)
160 ON WARNING NEXT :: DIM LD(12),C$(12),LM$(4)
170 MO$=" 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19
     20 21 22 23 24 25 26 27 28 29 30 31 "
180 GOTO 190 :: CAL$,D1,D$,E$,K,L,L$,L1,L2,L3,L4,L5,L6,M,N,P,S,
    W$,X,Y,YR :: CALL KEY :: !@P-
190 LD(1),LD(3),LD(5),LD(7),LD(8),LD(10),LD(12)=93 :: LD(4),LD(6),
    LD(9),LD(11)=90 :: LD(2)=84
200 DISPLAY AT(5,13)ERASE ALL:"PRINT": :S2$&" A": :S5$&S5$&"CALENDA
    R": :"from 1583 to the end of time": :RPT$(" ",28)
210 DISPLAY AT(16,8)BEEP:"OPEN PIO" :: ACCEPT AT(16,13)SIZE(-10):D$
    :: OPEN #1:D$ :: PRINT #1:E$&"@"
220 DISPLAY AT(16,1)BEEP 'Langu.: [E]nglish  [F]rench"
230 CALL KEY(0,K,S):: IF S=0 OR K<69 OR K>70) THEN 230 ELSE
    DISPLAY AT(16,9):CHR$(K):: IF K=69 THEN 270
240 LM$(1)=S2$&"JANVIER"&S8$&"FEVRIER"&S8$&" "&"MARS"
250 LM$(2)=S2$&" "&"AVRIL"&S8$&"   "&"MAI"&S8$&"   "&"JUIN" ::
    LM$(3)=S2$&"JUILLET"&S8$&" "&"AOUT"&S8$&" "&"SEPTEMBRE"
260 LM$(4)=S2$&"OCTOBRE"&S2$&S5$&"NOVEMBRE"&S2$&S5$&"DECEMBRE" ::
    W$=RPT$(S5$&" D  L  M  M  J  V  S",3):: GOTO 300
270 LM$(1)=S2$&"JANUARY"&S8$&"FEBRUARY"&S8$&"MARCH"
280 LM$(2)=S2$&" "&"APRIL"&S8$&"   "&"MAY"&S8$&"   "&"JUNE" ::
    LM$(3)=S2$&" "&"JULY"&S8$&"  "&"AUGUST"&S8$&"SEPTEMBER"
```

```
290 LM$(4)=S2$&"OCTOBER"&S2$&S5$&"NOVEMBER"&S2$&S5$&"DECEMBER" ::
    W$=RPT$(S5$&" S  M  T  W  T  F  S",3)
300 DISPLAY AT(19,1)BEEP:"from YEAR " :: ACCEPT AT(19,11):YR ::
    IF YR<1583 THEN 300
310 DISPLAY AT(19,17):"to "&STR$(YR):: ACCEPT AT(19,20)SIZE(-4)
    BEEP:S :: S=S-YR+1
315 ! ** calculate a year **
320 FOR Y=1 TO S :: DISPLAY AT(23,1):STR$(YR)&" / MONTH #"
330 FOR M=1 TO 12 :: DISPLAY AT(23,15):M
340 L1=YR-INT(0.6+(1/M)) :: L2=M+12*(INT(0.6+(1/M))):: 
    L3=INT(13*(L2+1)/5) :: L4=INT(5*L1/4):: L5=INT(L1/100)
350 L6=INT(L1/400) :: L=L3+L4+L6-L5 :: D1=L-7*INT(L/7)
360 CAL$=SEG$(MO$,1,LD(M)) :: IF M<>2 THEN 380
370 IF INT(YR/4)=YR/4 AND (INT(YR/100)<>YR/100 OR
    INT(YR/400)=YR/400) THEN CAL$=CAL$&" 29"
380 FOR X=1 TO D1 :: CAL$="   "&CAL$ :: NEXT X ::
    C$(M)=CAL$&RPT$(" ",126-LEN(CAL$)):: NEXT M
390 ! ** print a year **
400 DISPLAY AT(23,1):"PRINTING...";YR :: N=21 :: D$="      " ::
    PRINT #1:TAB(30);CHR$(14)&"~~";YR;"~~" :: FOR X=1 TO 4 ::
    P=1 :: M=3*X-2
410 PRINT #1:"":"":"":E$&"G"&LM$(X)&E$&"H":L$:E$&"G"&W$&E$&"H"
420 PRINT #1:S5$&SEG$(C$(M),P,N)&D$&SEG$(C$(M+1),P,N)&D$&
    SEG$(C$(M+2),P,N):: P=P+21 :: IF P<127 THEN 420
430 PRINT #1:L$ :: NEXT X :: PRINT #1:CHR$(12) :: YR=YR+1 :: NEXT Y
440 DISPLAY AT(22,1)BEEP:"CONT: [S]ame lang. [C]hange": :S5$&S5$&
    [Q]UIT PGM"
450 CALL KEY(0,K,S):: IF S=0 OR(K<>81 AND K<>83 AND K<>67) THEN 450
    ELSE IF K=81 THEN CLOSE #1 :: END
460 DISPLAY AT(22,1):"":"":"" :: IF K=83 THEN 300 ELSE 220
```

As usual, we start with initializing the variables, including some convenient spaces (5, 12 and 18 characters long); the L$ string will print a nice row of hyphens to frame each month's calendar (if your printer has graphics, just replace char. 45 with the appropriate character number for a horizontal line). The LD array is filled in line 190 with the position of the last day for the given month: January, March, etc. having 31 days, we get position 93 in the MO$ string (be careful when you type line 170: each day is three characters long, two for the day, plus an ending space).

After telling Tex the name of your printer (or DSKx.filename), you choose the language: it will then read the appropriate month-names and day-initials. (For the curious ones, the French days are Dimanche, Lundi, Mardi, Mercredi, Jeudi, Vendredi and Samedi). Finally, you are asked the starting and ending year; if you need only one year, just press enter to accept the default starting year; if you need more, enter the ending year, and Tex will know how many to print (how clever...!).

The program then really goes to work: for each month M of each year Y, it will build the CAL$ string. Now, once again, I am not sure I understand all the logic... this is the part I copied from Mr. Legault's program... Line 340 is the heart of the routine, properly calculating, I think, the day of the week on which the first of the month falls; more exactly, D1 is the number of empty spaces before the first of the month.

The string CAL$ now gets the proper portion of MO$ according to the position of the last day. If the month is February (M=2) and the year is a leap year (like 1988), the program adds a 29th day to CAL$. The leap year calculation works perfectly, even for century years like 1800, 1900 or 2000, which are leap years only if they divide by 400... this is checked by line 370. The D1 number of empty spaces is then added in front of the CAL$ string, and the whole thing is copied into the C$(M) array, with the right number of empty spaces for each month. Each C$(M) string has exactly 126 characters, so it is very easy to print it by segments corresponding to a week (21 char. x 6 weeks).

The printing portion is fairly straightforward. CHR$(14) in line 400 prints the year in shift-out mode, while line 410 uses the double-strike capabilities of Epson-compatible printers: "G" turns it on and "H" turns it off, E$ being the escape character (see line 150). This will emphasize the month names and day initials. A form feed, CHR$(12), is automatically printed after each year. When Tex has completed its task, you are asked to continue with the same language, the other one, or simply to exit the program.

TI BASIC continued from June
by Steven Shaw

QUICK SORT

Sorting data is a frequent task for many programs, and it seems reasonable to use the fastest sorting method available.

This sorting routine is very fast.

The variables used are:

A,B,C,D,E, F()
A$(), B$

To use the sort, your program must contain the initialisation lines at the beginning. You should not use the above variables in the main program, but if necessary you can change the variable names in this routine to avoid conflict.

The initialisation is here for 200 items. If you wish to sort a different number of items, set C to the number of items to be sorted (line 130) and DIMension A$ in line 100 to the number of items plus one. Then in line 150 set A$(Number of items plus one) to | (FCTN and A).

Your program may enter the routine with GOTO (EXIT with GOTO) or with GOSUB (EXIT with RETURN).

The items to be sorted are to be placed in the array A$(), and when the routine is finished, the items will still be in the array, but in ascending order,depending on the ASCII codes of their letters:

eg AA after A, B after AZZZ and so on. A$(0) is NOT used for the items to be sorted, it is a flag.

This routine will sort up to 1000 items. After that, you will need to DIMension the S array- to S(11) for 2000 items, S(12) for 4000 items and so on

```
 {quicksort}
Initialisation:
 100 DIM A$(201)
 110 A=1
 120 B=1
 130 C=200
 140 A$(0)=" "
 150 A$(201)="|"

    { program }
2000 IF C-B<10 THEN 2320
2010 D=B
2020 E=C
2030 B$=A$(B)
2040 IF B$>=A$(E)THEN 2070
2050 E=E-1
2060 GOTO 2040
2070 IF E>D THEN 2100
2080 A$(D)=B$
2090 GOTO 2190
2100 A$(D)=A$(E)
2110 D=D+1
2120 IF A$(D)<B$ THEN 2110
2130 IF E>D THEN 2170
2140 A$(E)=B$
2150 D=E
2160 GOTO 2190
2170 A$(E)=A$(D)
2180 GOTO 2050
2190 IF C-D<D-B THEN 2260
2200 F(A)=C
2210 A=A+1
2220 F(A)=D+1
2230 A=A+1  2240 C=D-1
2250 GOTO 2000
2260 F(A)=D-1
2270 A=A+1
2280 F(A)=B
```

```
2290 A=A+1
2300 B=D+1
2310 GOTO 2000
2320 E=B
2330 E=E+1
2340 IF E>C THEN 2430
2350 B$=A$(E)
2360 D=E-1
2370 IF A$(D)<=B$ THEN 2410
2380 A$(D+1)=A$(D)
2390 D=D-1
2400 GOTO 2370
2410 A$(D+1)=B$
2420 GOTO 2330
2430 IF A=1 THEN 2490
2440 A=A-1
2450 B=F(A)
2460 A=A-1
2470 C=F(A)
2480 GOTO 2000
2490 RETURN
```

EXTENDED BASIC

The Extended Basic module is not inexpensive, and information on what it can do is not widely available. Therefore this section has been included to help you to decide if you need the module, and to give some short hints on its use.

Extended Basic exists in two distinct versions, Vn 100 and Vn 110. Only a very small handful of the earlier Vn 100 have been sold in the UK, and this book refers to Vn 110. A principal difference is speed: 110 is much faster. There have also been changes in the operating systems which sometimes cause incompatability between the two versions (with special reference to the sprite routines and user sub programs).

EXTENDED BASIC is a VERY much larger language than TI Basic. The increase in operation speed is not shown by magazine 'bench tests' which use very short specific programs. In a typical program you will find the program runs in about 30% less time. Line transfers and screen handling are particularly faster than in TI Basic.

In itself, this is a great attraction, but Extended Basic also adds very many new commands and functions, enabling better use to be made of limited memory, and also permitting friendlier programs to be written.

Many TI BASIC programs can be loaded in Extended Basic, and will then run faster.
Exceptions are:
TI Basic programs over 12k cannot be loaded due to lack of memory.
Some TI Basic programs will load but cannot RUN due to lack of memory.
TI Basic has two extra character sets: if these are used, they will produce a BAD VALUE error in Extended Basic.Extended Basic uses the memory saved by dropping these sets (15 &16) to produce the Sprites.

What is new? The following is only a short list:

ACCEPT AT and DISPLAY AT permit you to display text anywhere on the screen, or to accept anywhere. There are numerous variations to these commands:
Optional 'beep',the ability to input data already on the screen or type over it,validation of input, and for the display of data, the ability to 'image' numbers. This allows simple justification of numbers.
Typical use: ACCEPT AT(6,12)BEEP VALIDATE("YN") SIZE(-1):A$

If there is a "Y" at row 6,column 12, just pressing ENTER will place Y into A$. Press any key but Y or N and the input will not be accepted, the computer returns to the line itself, and waits for a valid input.

CALL: In addition to the subprograms provided (eg COLOR, SOUND and so on) you may write your own sub programs, which you activate with CALL - for example CALL MYSUBPROG.

As with the TI subprograms, you may pass values or variables to your subprogram, and variables used in the subprogram are separate to variables in your main program. This is potentially a powerful programming capability.

Unfortunately the usual system error traps work badly with user written sub programs, and if an error message is generated it will usually be the wrong message for the error the computer has discovered. You must be careful how you write your subprogram!

A number of the built in CALLs have been extended, to allow you to define 4 characters at once, or amend all the colors at once- eg CALL COLOR(1,2,2,2,3,3,8,15,16) and so on.

This not only saves memory but also processes more quickly.

SPRITES you have probably heard of: they are smoothly moving graphics characters, which move under the control of the Video Display Processor, while your program carries on with other things.

TI allow you 28 sprites, each of one to four characters. They may also be double size (eg a 4 character sprite occupying the screen area of 16 characters).

There are subprograms built in to determine if sprites are in a particular position or overlapping, and you may quickly reposition them, change velocity, change colour or change the character of a sprite.

There are two restrictions which mean you need programming skill to use them to full effect:

The processor can only handle 4 sprites at a time in line. Each row of pixels is restricted to 4 sprites-any extra are made invisible.(A PIXEL is one dot in a character grid. A TI99/4A character is made up of dots in an 8 x 8 grid).

The CALL COINC coincidence checker only checks at the instant that command is used: you need to use it fairly often if a coincidence is not to be missed.

Nevertheless, in the hands of skilled (and patient) programmers, sprites can produce some VERY clever programs (in a form of the BASIC language too).

IF...THEN...ELSE has been greatly improved. You are no longer limited to line transfers, but may use:

```
IF A=4 AND B=6 THEN R=10 ELSE PRINT "OOPS"
```

Using IF..THEN..ELSE with commands enables you to use the memory available in a much better way.

LET

You do not actually use LET with the 99/4A, but it has to be listed somewhere...

In Extended Basic, instead of using: A=0 B=0 and so on,

you may assign one value to several variables in a neat and memory saving manner:

A,B,C,D,E,F,G=0 will reset all those variables to zero!

LINPUT: Ordinary INPUT removes leading spaces and causes problems if you wish to input a string with a comma in it. LINPUT avoids these problems. It stands for LINE INPUT.

LIST...when you LIST you may make the computer PAUSE in the list by pressing any key, then press another key to make the LIST continue.

ON BREAK NEXT disables the CLEAR key except when the computer has halted for an INPUT or ACCEPT AT. By avoiding these input commands you can make your program unbreakable..

ON ERROR is a VERY useful command. Normally an error causes your program to break, but you may use this command to transfer program execution to an error routine of your own. Your error routine may end by instructing the computer to try the problem line again, to go on to the next line, or to go to any other line in the program. You may also print your own error messages, such as "PRINTER NOT CONNECTED". This command can be used to make your programs totally user friendly. NB:Do not insert until your program is completely debugged!

ON WARNING is similar but with fewer options: you may halt the program or continue.

PROGRAM LINES: May now contain more than one command, and can be entered up to 5 screen lines long (but limited to 128 bytes long internally).

You may use IN COMMAND MODE for instance:

```
FOR A=110 to 220 :: CALL SOUND(200,A,0) :: NEXT A
```

The double colon is a statement separator. In TI Basic you could enter PRINT A::B::C.

In Extended Basic you must leave a space between the colons:

```
PRINT A: :B: :C
```

(A program in TI Basic is converted automatically by the machine to the new format, but you must take care when typing in a program. Due to an omission in the error handling system, typing too many colons together in Extended Basic can cause the processor to 'lock out')

When this is linked to the new capabilities of the IF...THEN command, it is possible to put together some very powerful program lines:

```
IF A=B  THEN C=5 :: PRINT A :: ELSE IF A=8 AND B=C THEN GOTO 3400 ELSE CALL
SOUND(100,110,0) :: GOTO 200
```

As the lines become longer and more complex, you do need to take greater care, but the language gives you a very powerful tool.

In addition to using REM after double colons, you may use a 'tail remark', which is a '!' as follows:

```
SCORE=0 ! RESET SCORE
```

RUN

It is possible to RUN one program from another:

```
RUN "CS1"  or  RUN "DSK1.PROGTWO"
```

SAVE: Programs may be saved in PROTECTED format, which prevents listing, editing or saving, and may be saved to DISK ONLY in Merge format, which allows program segments to be spliced together.

SPEECH:

From the speech editor module comes CALL SAY and SALL SPGET, which enable your Extended Basic program to use the speech synthesiser.Although TI provide a vocabulary list with the Extended Basic module, full instructions are not provided.

CALL SAY allows you to SAY a word from the vocabulary. Words NOT in the list will be spelt. Some 'words' are really phrases, but if you use CALL SAY("READY TO START"), the computer will SPELL the words! This is because the space is treated as a word separator. For the computer to recognise that these three words are one unit in the vocabulary, you need to enclose them in hash marks: CALL SAY("#READY TO START#")

The standard punctuation marks are also word separators and provide differing degrees of pause between words.

In order of length of pause, the separators are:

+ (space) - , ; : .

The + is zero pause and the full stop is a one second pause.

The separators may be repeated to build up any pause,eg:

```
CALL SAY("I---KNOW") or CALL SAY("I-,,KNOW")
```

CALL SPGET is used to fill a string variable with the data use by the speech synthesiser. This can increase the speed of execution if you fill some string variables at the start of your program, and then use them when you wish to speak. It takes a little while to fill the variables, as each string is 255 characters long (many of them are nul or zero value).

```
eg CALL SPGET("WORKING",WK$)
   then
   CALL SAY(,WK$) will say the word.
```

Note the comma in front of WK$. If two strings are used,in addition to the leading comma, they must be separated by TWO commas: CALL SAY(,A$,,B$)

In addition, you may use SEG$ to curtail the string you have returned. You can then separate the initial sounds of each word, and use these to create your own vocabulary. You are NOT limited to the preprogrammed list: you just have to work a little to expand it.

For example, having loaded WK$ as above, try:

```
WK$=SEG$(WK$,1,60)
CALL SAY(,WK$)
```

Notice any change? Try using different lengths in the SEG$ command. This is an area for experimentation.

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.

MICHAEL TAYLOR	PRESIDENT	831-0143
JANE LAFLAMME	VICE-PRESIDENT	(H) 837-1719 or (W) 745-2225
PETER ARPIN	TREASURER AND SYSOP	523-0017
JOHN O'CONNOR	SECRETARY	833-2626
BILL SPONCHIA	PAST PRESIDENT, SOFTWARE CONTEST	523-0878
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) 476-0265*
ART GREEN	ASSEMBLY HELP	837-1955
DICK PICHÉ	TECH	521-8667
CLUB BBS	SET MODEM TO 8N1	738-0617

Bob can be reached through his sister in North Bay at the number above. If you like, you can also reach him on Delphi (CDU) or on Compuserve (73657,3617). Watch for a more permanent number here.

LAFLAMME & WRIGLEY WHOLESALE
5480 Canotek Road, Unit #16
GLOUCESTER, ONTARO K1J 9H6
(613) 745-2225

GENEVE 9640

AUTHORIZED CANADIAN DISTRIBUTER FOR MYARC, INC.

SUPPORT YOUR CANADIAN DEALERS:

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

COMPUTER DOWNLOAD UNLIMITED
126 Sage Rd.
North Bay, Ontario
P1A 1A4
(705) 476-0265

DATA*PORT
2846 Gottingen St.
Halifax, N.S.
B3K 3E1
(902) 454-0232

DEALER ENQUIRIES WELCOME

OTTAWA, ONT.

Canada 57

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

EDMONTON 99er USER'S GROUP
P.O. BOX [illegible]
EDMONTON, ALBERTA
T5J 3L1