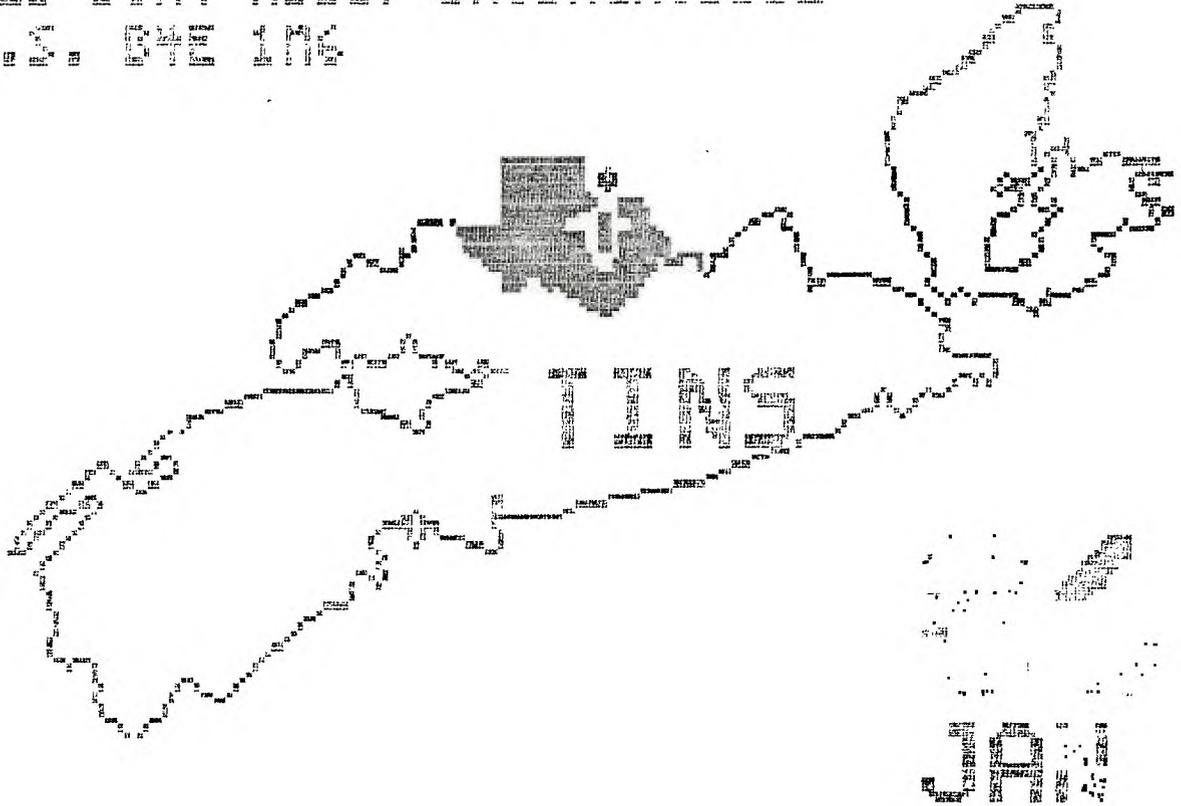


Sacku, N. S.
Jan 86

PROBATION BOARD

1000 W. 10th Avenue
Sackville, N.S.



MEMBERS

BARRY CONER	President	486	5345
LEW CURRIE	Worship	484	4526
KEVIN FLEMING	Secretary	485	3320

Editor

TINS Newsletter

TINS Newsletter is published on a monthly basis as the means of communicating ideas obtained from solicited sources to the general membership. Views expressed in this newsletter are those of the contributors and do not necessarily reflect the views of the membership at large.

Material appearing in this newsletter is copyrighted by compliance with federal regulations for basic copyright protection. The effective date of copyright is the 1st day of the month appearing on the edition.

Articles and programmes appearing in this newsletter are, to the best of our knowledge, original works except where indicated by the inclusion of the source. Arrangements have been made with other newsletter staffs and magazines for permission to reprint certain material. These articles are accepted on trust and the newsletter accepts no responsibility for searching the right-to-print of the originating periodical. Errors or omissions should be communicated to the editor as soon as possible.

Groups wishing to include material from these Newsletters in their club newsletters may do so providing the author and this source are mentioned. Other use of the material is subject to release by the editorial staff.

Space for advertising is available to merchants wishing to display their wares. Full page \$15, lesser sections at appropriately reduced rates (payable to TINS). All commercial ads must reach the editor in pre-prepared, ready to print form, not later than the 1st of the month in which the ad is to appear. Members may place ads in the Newsletter free of charge.

The Newsletter is on sale to members for \$1.00 per copy, non-members \$1.50. The price of each issue is solely to defray publication costs and does not represent profit. Prices for annual subscription are \$18.00.

Back issues of the Newsletter are available on written request from the editor, at \$1 per issue. First 3 disks of "The Best of TINS" on disk is available at \$3.00 per disk, contact editor.

All queries and newsletters should be forwarded to the address below, other correspondence should be directed to the Club at PO Box 3391, Dartmouth East, N.S. B2W 5G3.

Editor TINS Newsletter

321 Iony Hill
Lr. Sackville, NS
B4E 1M6

Other Correspondence:

TINS
Box 3391
Dartmouth, E., N.S.
B2W 5G3

The new year is here and we are looking forward to another twelve issues of the TINS newsletter. I sincerely hope that we will have a better year as far as INPUT to the newsletter is concerned. The past few months have been terrible.

In order to better define who is responsible for what, I have asked the following people of the executive for monthly notices or articles. Since these are the people who will be in the know, they should be able to keep you informed about what is happening in the club. These people are:

Barry Comer: Monthly Presidents message
Les Currie : Vice-Presidents report
Kevin Fleming: Whats in the mail bag
Librarians : Updates on the libraries.

These are not the only sources of articles for our newsletter. Anyone with an idea or a wish may write an article to please themselves. Remember, you are not the only person who is in the position your are. There are thousands of you, and each one of them would like to hear about what you have found or would be more than willing to help with a problem you have. The TI community is closer than any other computer community. Ask or share! Either way, write something for the newsletter.

Those with the expertise in the field that are members of the club have been contributing their knowledge to the newsletter on a regular basis. Thanks, and please keep up the interest and productivity. We need the expert opinion.

Plans for the new year, we have several articles that will be split over several issues. One of these appeared in the Nov and Dec issue and will continue until it is completed, sometime in the summer (ran out of room for it in this issue, sorry). That is the article on the hidden subroutines of the Personal Record Keeping and Statistics modules. Others include: Forth articles, Wy-Term review, review of the Jan TI-fest, equipment reviews and much more. An up to date library listing will be forthcoming as well as a newsletter directory.

If you have any things you would like to see included in the newsletter, let me know and we will see what we can do. If you don't bother to let me know or don't bother to write for your newsletter, then I will feel free to fill it up with articles from other groups. The more you input the more locally flavoured will the newsletter be.

Paul R. Meadows
Editor

President's Message

Well here we are another year behind us. There has been many changes in the last year that have brought our club closer together. There has been news of a new computer (which I hope comes our way in the near future), we have seen the leadership of the club turned over into my hands, we have seen our cartridge, disk, and tape libraries grow into a wealth of useful software for our club.

I feel that there has been one weak area in the club. This concerns everyone's participation in the NEWSLETTER. I strongly consider our newsletter to be the best newsletter that I have ever seen. We must have your articles to help keep it ours. Take a look through our letter and you will see certain names over and over again, or even worse names of people that do not even belong to our club. This is one subject that I will not let up on, even if I must confront everyone on a one to one basis I will. It is not Paul or Myself that need Your help, it is you that need your help.

I have seen other clubs including TI clubs in Ont. suffering from some members that are just along for a ride. You must remember that you pay a membership to belong to this club, it is up to you to get your money's worth. I remember when Tim was president of the club many moons ago, and I was one of the first members to join after he decided to open it up to the public. This club has come a very long way, but it still has a long way to go. I want it to be known that when other clubs see our newsletter, or talk to some people in our club that we know what we're doing.

One message that I want to pass on to you "WE need more input from you". If there is anything that I can do to help you with anything PLEASE ask. You have elected me (by default) to run this club, but I can't do it without your input. If you have anything that you would like to see done or changed let me know. In the future we will be having another program contest. I feel that the amount of entries in the last one was "VERY POOR". There is some very talented people in our club that did not even try to enter anything, I find this very disturbing. There is also many beginners in the club, you should not feel that your work is of any lower scale than that of someone that has been programming for years.

Membership in the club has been moderate to poor for the past year. 85 is the time to look at both renewing our old membership and bringing in more of those TI users who are still operating in the dark. With the possible rebirth of TI home computer users as a major participant in the computer community, more and more people will be looking for help. We will have to let them know we are around and that we will help all who

join us.

We have made an attempt to solve one of our major problems, this refers to the bringing in of a system every meeting. It was decided by the whole last exec. meeting that we will buy a system for the club. Basic computers in Halifax had an expansion system for sale, it was decided that we will buy it for \$200.00. It contains a RS232, Disk card, and 1 disk drive. We need a memory card, so if you here of anyone that has one for sale let us know. The system will not be lent out because the system is very fragile. I do however want to run an adventure board from this system, let me know if it would be worth while.

In short I want you to know that I along with Les Currie your new Vice Prez. will do what we can to keep up the high standards that have been set down by the people before us.



Cassette Library

Sheila Dickson

There seems to be little interest shown in the cassette library of late. Has everyone gotten disk drives now? There have been three more copies of "Teach Yourself Extended Basic" donated to our library. Other recent additions are:

- CCC Rally
- Graphics Pkg.
- Vocabulary Quiz
- Selftest on TI99/4A
- 99'er Programs
- Home Computer Programs

A revised list of cassettes is available. Please see either Les Currie or Sheila Dickson.

SORTING

John Clulow

This article was downloaded from TIBBS and I believe that the author is as stated above. The short basic programme following gives an excellent demo of what the sort routine can do.

CODE

```
00CA0      A000A07D0A08D409D0A0AD0A0AF0B0000B0001BFF00BC80B7F2ADF      0001
A0AFBC0AF0B02E0C0AD0B04C0B0201B0002B0420B200CB0200B4041B90207F2D1F      0002
A0B0EB834AB1316B06C0B9020B834AB1304B0200B1300B0420B2034B02027F2FBF      0003
A0B24B0064B04C3B04C4B0E0B834B006C3B38C2B04C3B00E0B834CB06C37F254F      0004
A0B3ABA103B100404C4BD120B834B006C4B04CAB04C9BC820CAF2C00027F270F      0005
A0B50BC804C0004B058AB028AB0000B1602B0460C0C58B060ABC24AB0A197F2AEF      0006
A0B66BC049B0A11BC321C0002BC361C00045C00CB0201B0001B0202C07D07F2D9F      0007
A0B7CB0820C0AF4C07D0B0420B2014BC3CCBC38DB058EB060EB83CEB133A7F21DF      0008
A0B92BC00EB0201B0001B0202C08D08D0820C0AF4C08D0B0420B2014B02017F2D1F      0009
A0BABC07D0B0202C08D0B06A0C0C65B0280B0001B1301B10E9BC00FB02017F2B6F      0010
A0BBEB0001B0202C08D0B0420B2010B058FB83CEB131CBC00FB0201B00017F2B3F      0011
A0BD4B0202C09D0BD820CAF4C09D0B0420B2014B0201C07D0B0202C09D07F2B0F      0012
A0BEA506A0C0C65B0280B0002B1301B10E9BC00EB0201B0001B0202C09D07F2C3F      0013
A0C00B0420B2010B10C3BC00FB0201B0001B0202C07D0B0420B2010BC04D7F2FEF      0014
A0C16B604FB0281B0002B110BBC24AB0A19BC04FB0581BC089B0A12BC8B17F27BF      0015
A0C2CC0002BC88DC0004B058ABC04EB0604CB0281B0002B110BBC24AB0A197F27EF      0016
A0C42BC089B0A12BC88CC0002BC04EB0601BC8B1C0004B058AB0460C0B567F286F      0017
A0C58B04C08C800B837CB02E0B83E0BC2E0CAF0B045BB04C3B00D1BC0027F23FF      0018
A0C6EB9452B1401B0D02B06C3B0581B0582B9452B1A0BB1B0DB0603B15F97F27AF      0019
A0C84BC080B9812C07D0B1A04B1B06B0200B0003B045B0200B0001B045B7F2DBF      0020
A0C9AB0200B0002B045B7F7B76F      0021
50AF65SORT 7FD1FF      0022
:          99/4 AS      0023
```

BASIC Programme

```
100 CALL INIT
110 CALL LOAD("DSK1.SORT")
115 OPTION BASE 1
120 DIM A$(20)
130 CALL CLEAR :: FOR I=4 TO 19 :: READ X$ :: DISPLAY AT(I,1):X$ :: NEXT I :: DI
SPLAY AT(23,2):"PRESS ANY KEY TO CONTINUE"
140 CALL KEY(0,K,S):: IF S=0 THEN 140
150 DISPLAY AT(1,3)ERASE ALL:"Here's a random array:"
160 FOR I=1 TO 20 :: FOR J=1 TO INT(RND*5+5):: A$(I)=A$(I)&CHR$(RND+65):: NE
T J :: DISPLAY AT(I+2,2):A$(I):: NEXT I
170 DISPLAY AT(24,1)BEEP:"PRESS ANY KEY TO SORT..."
180 CALL KEY(0,K,S):: IF S=0 THEN 180
190 DISPLAY AT(24,1):"  SORTING..."
200 CALL LINK("SORT",A$( ),20)
210 FOR I=1 TO 20 :: DISPLAY AT(I+2,16):A$(I):: NEXT I
220 DISPLAY AT(24,1):"  PRESS ANY KEY TO QUIT"
230 CALL KEY(0,K,S):: IF S=0 THEN 230
235 RUN "DSK1.LOAD"
240 DATA "SORT does an ASCII quick-","sort on any string array.,""The Ext BASIC
program should"

250 DATA "use OPTION BASE 1.,""Load the subprogram with...","CALL INIT","CALL
LOAD("DSK1.SORT")"
260 DATA ,"To sort an array X$( ), use","CALL LINK("SORT",X$( ),X)","where X is
the last element","in the array."
270 DATA "The sorted array is returned","in X$( )."
```



Out and Around

Editor

As some of you are aware, I had the opportunity of visiting the Edmonton TI User Group meeting instead of attending our own monthly meeting last month.

I must say that I was overwhelmed by the courtesy and help that I received. I would like to thank all of the club members for making the visit a pleasant and interesting one. Special thanks to the two individuals who went so far out of their way to pick me up and return me to my quarters. Hope we can return the favour one of these days.

For those who are anxiously awaiting my comments on how the Edmonton Club compares to TINS, sorry! The monthly meeting went on much as do our own. It seems that TIers are the same all over. A group of TI enthusiasts gathered in a room on the University Campus and did all of the same things that we do at our meeting. There were discussions of FORTH, BASIC, "C" and other languages, clamouring for newsletters, sale of some equipment and library activity. Sound familiar! The keynote of the evening was a talk and demo on the ins and outs of TI-Writer by Bob Pass. Even an old TI-writer enthusiast (a bias one too) found food for thought in the talk. Thanks Bob.

Later in the week, I had the opportunity to visit with Tom Hall and Michal Jaegermann at Tom's apartment. I had brought some disks with me containing various Freeware programmes and some of my own to pass along. Talk about embarrassing happenings, all the disks had been zapped and proved unrunnable! Oh well, thats the hazards.

Notwithstanding my problem, Michal gave me a very interesting demo of his new "POWER" editor for TI-FORTH and provided me with a copy for our library at the same time. I was so impressed with this editor that I have taken space to include the bulk of a letter that Michal sent to LCT (author of the original editor). Get your copy from the TINS library or copy these screens.

POWER Editor Michal Jaegermann

In this article you will find a new editor for TI-Forth. But why bother if the existing editor is a quite decent one? Well, you will find in this new editor a lot of useful features which will make it into a quite a powerful tool which will assist you not only in creation but also in debugging your Forth programmes.

First of all it sports an auto-repeating keyboard, which is useful by itself. But you will find also

overtime and insert modes and a limited but very convenient form of "cut-and-paste". On the top of it you will find an ability to single step through your source screen, with a continuous stack display and a possibility to execute any Forth word without leaving the editor. This last feature is a "real Forth" in the sense that you have not only full control over results of your actions, but also full responsibility. So be careful! In addition, this new editor adds to the system less than 2K of editor specific compiled code. So it is approximately of the same size as the old editor. Too good to be true? Read on!

One more thing - while writing this editor I tried to be as compatible, with the old one, as possible. So you do not have to "unlearn" very few old habits in order to switch. Hope that this sounds attractive.

If you are still wondering about which editor, of the two supplied, I am talking about the answer is "both". You will find that the same editor is used in both modes with two extra screens taking care of mode dependent display details and minimal differences in compilation of a couple of words.

You will find the code in screens following this article. (Editor's note: screens will be displayed over the next few newsletters to conserve space. Save typing and get a copy from the library)

Here is a description on how to use the new editor.

System Requirements

First, the bad news. You need some extra words to compile the editor.

Good news - not too much and really handy on its own. Actually only one is really necessary: CMOVE) - move up memory contents. It requires a starting address on the stack, a target address and a count in bytes. It will not do anything if a count is not positive. For speed reasons definition is in code.

```
HEX CODE CMOVE) C0B9 , C079 , C039 , A002 ,  
A042 , 0600 , 0601 , 0602 ,  
1102 , D450 , 10FA , 045F ,
```

You will also need a .S for a stack display. It loads as one of the -DUMP words but you may extract its definition from somewhere else if you wish. Three others are "convenient" words - if you do not like them - edit the source and forget about them.

They are:
: \ IN C/L + C/L MINUS AND IN ! ; IMMEDIATE

: 2DUP OVER OVER ; (quite obvious)

```
: AT GOTGXY ( I am
allergic
to GOTDXY)
```

About the 64 column display. Since I have problems with an overscan, I moved the 64 column editor screen to the right. This is done by modification of SMASH on screen 65 of the system disk. In this code you will find (once only!) an entry 2000, . Replace it with 200B, . This will cause end-of-lines to disappear, but I think that this is a smaller problem. If you are lucky enough not to have an overscan problem then leave SMASH alone but remove 2+ from 64-column .CUR. by the way: CLIST (n -) (list contents of a block n) may be defined as:

```
: CLIST ( n - ) BLOCK L/SCR 0
DD I C/L * OVER + C/L I SMASH VMBW LOOP DROP ;
and you do not really need CLINE and CLOOP.
```

The editor will trap all non-printable characters with one exception DEL, (hex 7f). If this is going to bother you add in the following in EDI loop, right after RKEY:

```
' DUP 7F ( * '. This will remove the problem.
```

If you find that the sensitivity of a keyboard does not suit your taste - play around with a delay loop in BLINK and constants imbedded in RKEY. They are not exactly independent but try it yourself.

After you have done all of the above you may load your new editor and try to use it now.

Starting and Leaving Editor

As usual, 20 EDIT will bring the contents of block 20 onto your screen with an edit cursor in a home position. EDX will work also as usual. WHERE brings you to a location of a LOAD error.

One extra - ER (EditResume) recalls not only the last screen but also the last cursor position. So you will be back where you left the editor the last time. Once in the editor ctrl-E will switch to the previous screen (at home position), and ctrl-X to the next one. Fctn-9 to get out.

Entering Text

Editor will come up in an otype mode. So whatever you are typing replaces the text under cursor. Fctn-2 toggles between otype and an insert mode. While inserting a new text, the old text is pushed to the right. Whatever spills over the right margin is lost.

Marking and Unmarking Text

Think of it this way. You have always exactly two

marks. If they are invisible, then they are at the end of the current line and at your cursor position. Ctrl-Z puts a visible mark where your cursor is. The first one - end of line. If you will try to put a third mark on a screen - the second one will be replaced with a new one. Visible marks are stored on stack, so if you have to move the first one, SWAP them (how to do it later). Ctrl-U will erase all visible marks from your screen.

Deleting and Inserting In One Line

Fctn - 1 will delete one character. Remember - it auto repeats.

Fctn - 3 deletes the whole current line and all subsequent text moves up. Deleted line is stored in a delete buffer.

Ctrl - 8 opens a blank line over the cursor. Old last line is lost. So everything is as one would expect.

Fctn - 7 removes all text between marks (visible or invisible) and replaces it with blanks. Up to 64 characters of removed text are stored in a delete buffer. If there are more - they are lost.

The action of Fctn-8 depends on the editor mode. In otype mode it acts as Ctrl-8 but it moves text from the delete buffer into an opened line. While in insert mode - it inserts text from the delete buffer, without leading and trailing (but one) blanks in line, on the cursor position. Old text moves to the right. The right margin spillover is lost. Reread section on marking and experiment until you feel comfortable.

Do not hold Fctn-7 too long, since it will clobber your delete buffer with freshly created blanks.

Additionally, you may yank text to the delete buffer with Ctrl-Y. This will put away text between marks for subsequent Fctn-8 inserting without removing the original from a screen. The same limitations as above will apply here.

Moving Around

Usual "arrow keys" will work (Fctn S,X,E,D). But you also have "terminal style" controls (Ctrl-H=left, Ctrl-J=down, Ctrl-K=up, Ctrl-L =right). They are handy when single stepping through a source. Moreover Ctrl-R will move to the right in one word steps. No key for a similar movement backwards.

Single-step and Executing Forth

Ctrl-W will execute (if possible) any word on your

screen which is pointed by the cursor. The cursor will advance to the next word and you will see a display of the stack below your edit screen. Great for debugging.

Ctrl-Q will do the same with two words (try @ CLOAD EDIT). Do not try to execute compiling words like DO or IF since you will get an error. No big deal, ER will put you where you have just been, but the stack will be lost.

You may also hit Ctrl-. (control period) to run an internal interpreter. You will be out below the edit screen and you may type there up to 80 characters of Forth to execute. Upon enter you will return, automatically, to the editor and the current stack will be displayed below. The editor part of the screen is frozen and will not scroll even if you dump the whole memory. This means that if you make an error then to unfreeze the screen, you have to return to editor (ER is ok) to get out later with Fctr-9.

This interpreter is even flexible enough to start a compilation (say with a colon definition), to return to editor, to do some editing and to resume a suspended compilation later. I am not advising you to use it normally in that way, but this is a great way to see for yourself how the compiler security is implemented and what DO is putting on stack to tell LOOP where to branch.

You will find out that in particular you may, using Ctrl-., call editor itself - executing, for example, EDIT or ER. I would advise you not to do that. Reason is that you are storing on a return stack a return address. So, once you would like to get out and hit Fctr-9 you will return... back to editor (previous instance). If you do that many time getting back to FORT may take a number of Fctr-9s. Ctrl-. and QUIT will always save the day.

How to move big blocks

The editor above, of course, can be extended and made more powerful. But a goal was to make it convenient, nice and not too big. For example, one may add a big delete buffer and rewrite deleting and inserting a little bit to get a full "cut-and-paste". But instead of doing this I am using the Ctrl-. feature on those infrequent occasions when I need more extensive capabilities. For example - how to move a block of five lines from screen number 23 into some other location on screen 37. Type 23 EDIT. In the editor, hit Ctrl-. and once outside EDITOR #CUR. The word #CUR from the EDITOR vocabulary returns an address in an edit buffer which corresponds to a current cursor position. You will see it on stack. Now Ctrl-9 and 37 EDIT. Once back in the editor at the very beginning of an imported block. To mark the block as an update, just retype one

character on screen. FLUSH will save all changes to disk. Another... For such operations, from the EDITOR vocabulary... ROOM. See source screen for details

How to Save Typing

For out of town readers:

Send a disk in a self-addressed mailer (with proper postage included). If not in Canada you may buy a "International Reply Coupon" in your post office. You will get back a modified TI-Forth system disk with this editor and many other handy utilities included. It is really worth it. The address is:

Edmonton 99'er Computer User's Society
Box 11983, Edmonton, Alberta
Canada T6J-3L1

For TINS members: see Librarian!

A Short Reference Guide

Function keys:

- 1 delete character
- 2 toggle overtyping/insert mode
- 3 delete line
- 5 swap windows in text mode/home in bitmap mode
- 6 move right
- 7 delete between marks (one or both may be defaults)
- 8 insert text from PAD
 - * in overtyping mode inserts a new line moving other text down
 - * in insert mode inserts content of PAD on the cursor position shifting text on the line to the right
- 9 leave editor
- S, X, E, D move cursor to left, right, up and down

Control keys

- B open blank line
 - E get previous screen
 - X get next screen
 - W execute one word pointed by cursor - display stack
 - Q execute two words pointed by cursor - display stack
 - R move right one word
 - Y yank - store text between marks (64 max) in edit buffer
 - Z mark cursor position
 - U unmark - replace marks by defaults (cursor, end of line)
- Forth available. After (enter) or 80 characters typed returns automatically to the editor with a display of the stack. If you get an ERROR type ER to return back to the editor. Otherwise an edit screen will be frozen.
- H, J, K, L move left, down, up, right - terminal style

Support Your TI

Another Fest

Barry Coover

LA 99er

In the interest of keeping you up-to-date on whats happening on the firing-lines of TI home computing, I have included this news received from California.

(Downloaded from TIBBS) Editor

LA 99er CUG announces the first convention and seminars for 99/4A owners and users. This will be a 2 day event scheduled for March 1 and 2 in Los Angeles. Included will be family type outings such as Disneyland and Universal Studio. We will have a hospitality suite open the entire morning Friday Feb. 28 through Sun. on 2 arriving and late leaving. If early indicates an adequate attendance. I'll attempt to lease an entire Motel, and consecrating the per bed rate down appreciably. We will attempt to book an entire floor. We plan a facility with meeting banquet facilities and will have a social period of about 1/2 hours following close of the day, followed by a Dinner for all, local groups are economy minded and will do all possible to keep the overall package price low. We will also take advantage of group or special air fares, 6% and for those interested Rental Cars. We will work life in the Travel Industry, to handle this end. Invited participants will be many of the well known names vital to our onward success. Vendor costs are being estimated now and will be forwarded shortly to them shortly by mail. All SOFTWARE programmers will receive an invitation to participate, we will encourage them to let the community see what wonderful, imaginative, vital persons they are. We invite you to publish this information in local newsletters and on local BBS. Please contact me via CIS or US MAIL to 148 S. Maple Drive, Beverly Hills, Ca 90212, or by phone at 213 2716930. Thanks and hope to see you here in March. The East Coast will soon sponsor the next one, most likely by the wonderful Philadelphia group, who gave us the fantastic "Mack" seminar. Watch for that announcement too. Thanks, Teresa Masters

It seems that it was only yesterday that we were wishing everyone the best of wishes for 1985. You reflect sometimes over the year past and ask what are some really good things that I have done for myself, and others.

I myself can think of many things that I have done in the last year that have helped me in the never ending learning of the computer field. I have learned many new things about how my machine works, I've written some articles for the newsletter, I even got some of those articles published in "Micropendium".

There has been great response to my release of the "Forth-Draw" program, there has been new friends made in other parts of the world. This brings me to the purpose of this article, the support that I have seen in the TI world has been nothing short of remarkable.

There is a very good example of this support in our very own club, there seems to be a bond between some members of our club. I have seen some people go out of their way to make sure that others have their questions answered.

In the last year we have had news of a new computer, which has been the promise of new life to our orphan machine. Well I say that this little orphan has seen the year out very well. When you compare the TI to other machines that have traveled the same highway we have held our own above all.

With some of our strongest support coming from companies like Corcomp, and Myarc we can not help but to keep faith in our machines. I have written some programs for the "Freeware" concept, the return has not been that I could go at it one hundred percent, but I feel that if everyone gives a bit we get alot.

In closing I want to say thank you to all of those people that have given to the cause, and future thanks to those who help us out in the new year.

TI Drums

```
100 CALL INIT::CALL SOUND(100,110,0)::FOR =1 TO 5
110 FOR A=1 TO 975 STEP 4::CALL LOAD(-31744,A)::NEXT A::FOR A=1 TO 255
STEP -5::CALL LOAD(-31744,A)::NEXT A::NEXT
```

WIERD

```
100 CALL INIT::CALL CLEAR::CALL MAGNIFY(4)::FOR B=1 TO 10::FOR A=100
TO 255::PRINT A::CALL LOAD(-31873,-A)::PRINT CHR$(A)::NEXT A::NEXT B::GC
```

TeleCommunication

ACCESS

The following was downloaded from ACCESS which is a large database on Datapac. The article is self explanatory.

In order to get connected to ACCESS you must first establish contact with Datapac. In Nova Scotia that is accomplished by phoning 477-2000 and then entering a "." if you are using 300 Baud. 1200 Baud is obtained by calling 477-8000.

On establishing contact with Datapac (you will get a Datapac port connection message) enter 49700019. This will connect you to Access. You will not see anything happen on the screen, since with service does not issue a sign-on banner at this point. Enter a carriage return ((enter)) and you will see a colon appear at the lefthand edge of your screen. Enter "HELLO DEMO.DEMO" at this point and the text below will begin to scroll.

```
:HELLO DEMO.DEMO
*** Welcome to Access ***
    MPE IV C.B1.A2
SAT, JAN 4, 1986, 12:26 PM
```

```
Last Call:
SAT, JAN 4, 1986, 12:20 PM
```

How many characters can your terminal print on one line? __

We are proud to welcome YOU to Canada's Online Community, ACCESS!

(You can pause this demonstration by entering a Control S and resume it by entering a Control Q)

You are probably wondering just what Access is. The easiest way to put it would be to say that Access is an electronic community. A community created by the phone line, the modem, and YOUR computer. But more than that, Access is a place where it is extremely easy to communicate. A simple 'send' command allows you to send messages instantly to any other person on the system. In our Comune, all can gather for informal chat, or roundtable discussions. Our Mail system never sleeps. It's open 24hrs a day, from coast to coast via Datapac.

ACCESS - for the Hobbyist AND the Business Community!

For the Hobbyist, we have multi-user real-time games like Spaxx, a Space Dogfight Simulation, Special Interest Groups (SIGS) for MSDOS, CBM, ATARI, CP/ M,

TRSEIGHTY SERIES, TRS MODEL 100 and APPLE...plus many, many more! There is a Communal Novel in the works, with participants adding chapters every so often, a Story Board, where ACCESSories (that's what we call ourselves) write bulletins, creating the storyline along the way. ACHMED, (A Chess MEDIator) will let you play against any other user, or MANY at the same time. Are you a gamer? ACCESS has just added a GAMING SIG for those interested in reading game reviews and discussing everything about gaming. The Hobbyist is well catered-to at ACCESS.

For more serious applications, not only do we offer online programming in 7 languages (including Pascal and Basic), but ACCESS offers you a National Electronic Mail Service at Canadian rates. Business users may wish to expand their use of the system and take advantage of our 'Online Office' Package, available at a low monthly fee.

ACCESS was designed from the ground up to be fun AND informative. Most other online systems are outgrowths of business information services. ACCESS is for the hobbyist, while certain sections have been set up specifically for business use - you have the best of BOTH worlds!

Easy help menus guide you on your way through ACCESS, and within very little time you'll be whizzing through at Highway speed. Your time on ACCESS is measured in minutes, so you always know precisely how much time you use.

ACCESS is a truly National system. 100% Canadian, we charge Canadian prices! Not only do you SAVE money by joining ACCESS, you also get to chat with people from all across the country about things that are important to YOU! (*Flash! Access has just joined "Inet 2000"...Canada's Gateway network!*)

Now let's get down to brass tacks: How much does it cost me?

TOTAL cost (INCLUDING DATAPAC) from anywhere in Canada, 24 hours a day, 7 days a week, 300 OR 1200 baud:

\$ 5 . 9 5 Per Hour !!

(Compare OUR prices to those of the large American systems)

Your Registration Pack is just \$45.00. This is not a yearly fee, but a one-time initiation fee. Your Registration Pack includes our Coil-bound User Manual (for easy desktop viewing), 4 hours of connect time, a password slip, and the Access News Update Letter, so you can keep abreast of what's

happening NOW on Canada's Fastest Moving Multiuser System!

Here is a map of ACCESS. Our map displays ACCESS's different areas. Keep in mi that DATABANKS are text databases, and the WALLS are Public Message Bases.

(Control S to pause Control Q to restart Control Y to abort)

The ACCESS Timesharing System

BUSINESS USERS

BUSPROGS..... Handy Business programs

For more Information on Business programs and Special Sub-ACCESSes, please contact us via Voice Call at (514) 342-8147.

COMM Communications
 COMMUNE Multi-User
 Teleconferencing
 MAIL ACCESS Mail Service
 SURVEY ACCESS Survey Poll Board
 WALL ACCESS Bulletin Wall
 DATA INFORMATION AREA
 DATABANK ACCESS Information Center
 UPDOWN Upload/Download Service
 EDUCATE Educational Programs
 BANKSIM Banking Simulation
 CARDIO Cardiovascular Simulation
 STOCKSIM Stock Market Simulation
 VOCAB Learning of Synonyms
 GROUPS SPECIAL INTEREST GROUPS
 COMPUTER COMPUTER SIGS
 APPLE APPLE User's Group
 DATABANK . APPLE Information Center
 SURVEY ... APPLE Survey polling Board
 UPDOWN ... Upload/download Service
 WALL APPLE SIG Bulletin Board

The Following Sigs also have DATABANK, SURVEY, UPDOWN AND A WALL of their own:

ATARI Atari User's Group
 CBM Commodore User's Group
 CPM CPM User's Group
 MSDOS MSDOS User's Group
 TRSEIGHT TRS-80 User's Group
 TRSPOR Tandy Model 100/200 User's group

With more Sigs being created all the time!

DEVELOP ACCESS DEVELOPMENT AREA
 PROGRAM Programming SIG
 DATABANK . Programming Information

Center
 WALL Programming Bulletin Board
 UTILS HP3000 Utilities
 DBUTIL ... DBUTIL/3000 - Database Utility
 EDITOR ... EDIT/3000 - HP Text Editor
 FCOPY FCOPY/3000 - HP File Copier
 QUERY QUERY/3000 - Database Inquiry
 RESIZE ... Shrink Text Files Down To Size
 UNITOP ... Universal Top Score Program
 UNITOPFM . File Formatter for UNITOP
 GAMING Adventure And Simulation Gaming
 DATABANK Gaming Information Center
 SURVEY Gaming Survey Polling Board
 WALL Gaming SIG Bulletin Board
 GEOGRAPH Regional SIGS
 MONTREAL Montreal Regional SIG
 TORONTO Toronto Regional SIG
 (With Vancouver, Winnipeg and Regina Soon to come!)

HUG HANDICAP USER'S GROUP
 COMMUNE Teleconferencing
 HEARING Hearing Impaired Deaf
 LEARNING Learning Disabled
 MENTAL Mentally Retarded
 NEWS HUG News Update
 PHYSICAL Physically Disabled
 SPEECH Speech Disabled
 VISION Vision Impaired Blind
 WALL HUG Bulletin Wall
 TEENS TEENAGE SIG
 INFO ACCESS Information And *help*

DIRECTRY Your Personal File
 Directory On ACCESS
 IDEABOX Submit ACCESS Suggestions
 LEISURE Entertainment
 GAMES Single-Player Games Sub-ACCESS

DRAGRACE Quarter-Mile Drag Race
 ELIZA Talk Over Your Problems With Eliza
 FOOTBALL American Football Simulation
 GOLF The Great ACCESS Golf Course
 KRUNCH Who Has Crashed In Rallye
 LOTO Generate Loto (6/49) Numbers
 MANSION Mystery mansion Adventure
 MILLBORN Race to 1000 Miles!
 RALLYE Vector-Based Car racing

game!

(Plus Many, Many More, too numerous to mention!)

- MULTIGAM MULTI-PLAYER GAMES
- ACHMED ACCESS Canada-Wide Chess League!
- JOEZBAR Multi-Game Bulletin Board
- OTTOBOOT Start Ottomatic Spaxx Robot Ship!
- SPAXX Space Dogfight..Pilot Your OWN Ship!
- SPREGAME SPAXX Create Your Own Ship!
- SPXTUTOR SPAXX Tutor..Learn How to Play Spaxx
- TANK Multi-Player Tank War Game!
- WARLDRS Multi-Player Medieval Combat!
- MUSIC MUSIC PROGRAMS
- MISC MISCELLANEOUS PROGRAMS
- ORACLE The Mystic Oracle
- STORY Story-Round Tall tales

ACCESS is a Flexible System. Area's and Sections are added all the time! As user interest warrants, you can start YOUR own SIG too! Belong to a group of any sort? (Computers, cooking, video's, cars etc) Access is the IDEAL place to get National Exposure! And we'll help you EVERY step of the way...even to the extent of creating your OWN group's area on Access, creating a group account, and plunking in FREE hours every month!

We'd like YOU to join us on Access! Remember, Access is only \$5.95 per hour from ANYWHERE in Canada, and that's a TOTAL cost, including long distance charge when Canada Post brings your Registration Pack to your door, inside you'll find your Access User manual, your passwords, your credit for 4 hours of Connecttime, and our Access News Update Letter, keeping you abreast of the many NEW events here at Access. All you have to do is open up your Pack, read it through, and DIAL!

Come on in....there's a whole new world out there just waiting to be discovered! A C C E S S !!

Here's how to join:

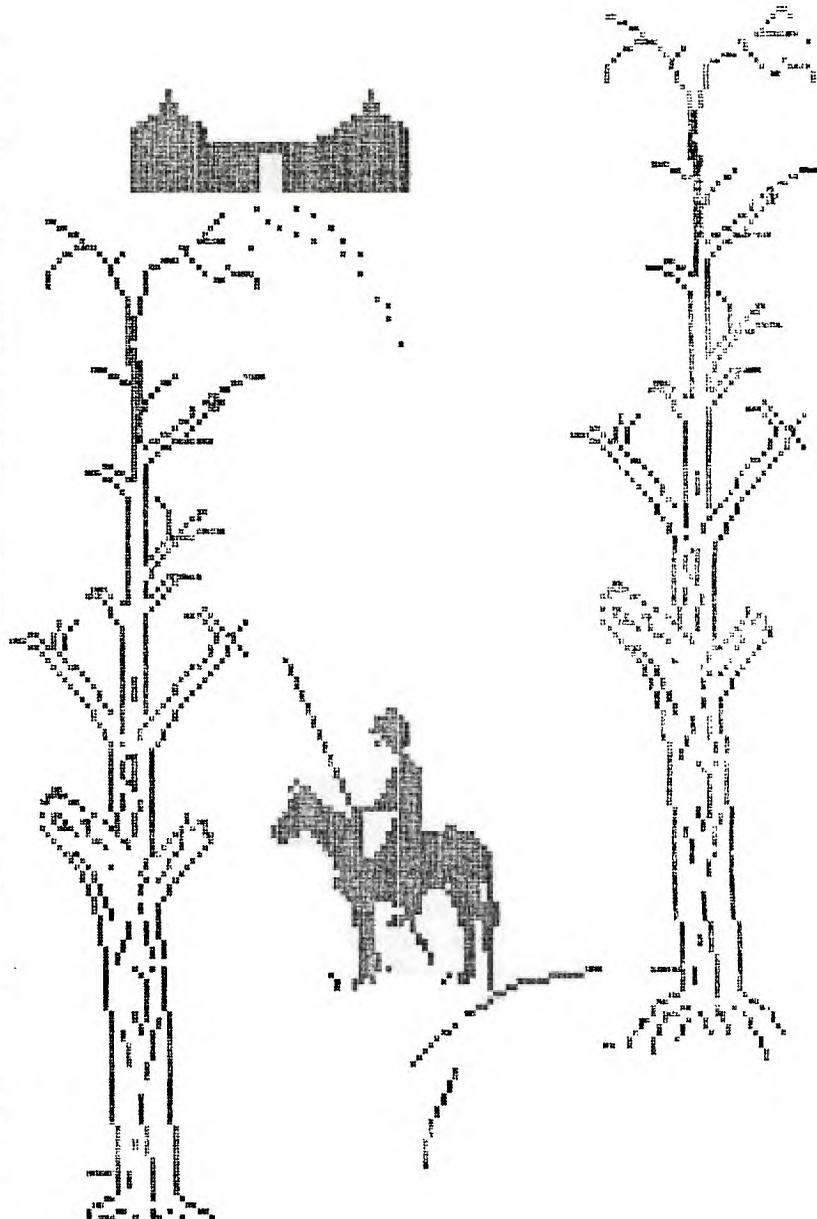
- (1) Credit Card - RUSH! I want ACCESS within DAYS!
- (2) C.O.D. (No extra mailing charges)
- (3) I'll think about it and call back when I'm ready.

Remember: your Credit Card will insure IMMEDIATE processing and shipping.

Please enter your option:

Choice: 3

This system has some of the features of TIMELINE as well as some that are new. If you take the opportunity to call and register as a member, let us know how you feel about the system. Perhaps a short article wouldn't been too much to ask. Since I am running a board of my own, I don't have the opportunity to get out and around the systems as much as I would like to, so I will have to rely on you to let me know. **Editor**



SQUAWKING

Tim MacEachern

RS232 Telecommunications Formats

This article describes the form of characters as they are transmitted when using modems via an RS232 interface. The intention of the article is to allow BBS users to understand what the basic settings actually control. If this can be achieved BBS users will be able to diagnose and correct communications problems more easily.

First I will describe the manner in which simple ASCII characters are sent from a computer to a modem (and vice-versa). Note that I am not describing the method that these characters are transmitted from one modem to another.

Let us consider transmission of the letter 'B'. Using the ASCII character set, defined characters are stored as numbers from 0 to 127 (decimal). 'B' is decimal 66 or hexadecimal >42 or, in binary, 100 0010. Note that the binary representation has only 7 bits, rather than 8. The ASCII character set comprises only 7 bit characters.

When the letter 'B' is sent via an asynchronous RS232 connection the transmission consists of four parts: a start bit, the data bits (sent least significant bits first), an optional parity bit and one or two stop bits. The simplest setup is 7 data bits/no parity/one stop bit. In this case a letter would be sent as binary bits 'sdddddde' where s is the start bit and e is a stop bit.

An RS232 line can be thought of as being at a '1' level when no characters are being transmitted. The start bit is always 0, signalling that good things are following. Next, the data bits follow, in the order least significant bit first, most significant bit last. After the data bits there may be a parity bit. Finally, there are one or two stop bits, which are always at the '1' level, the same as the idle state. This bit (usually there is only one) gives the receiver time to get ready for the next asynchronously received character. (Asynchronous means that it could come at any time after the last one). Lets get back to the example. Given the settings are 7/none/1 (data bits/parity/stop bits) then we expect to receive a string of characters 'sdddddde'. On an otherwise idle line a letter B (42 hex) would look like:

'...1111001000111111111...'
meaning
'...iiiiisdddddeiiiiiiiiiii...'

I'm using 'i' to represent the idle line state, 's' to represent the start bit, 'd' to show the data bits and 'e' to represent the stop bit(s). When parity bits are added they will be shown as 'p'. The receiver waits on the idle line (at level 1) until it gets a 0 (telling it to wake up!). It then reads the data bits (0100010). The stop bit is used as both padding to separate this from the next character and as a smooth transition back to idle. If the stop bit is not there (that is, the line is at level 0 when it is supposed to be at level 1) the receiver will complain about a FRAMING ERROR. To get the B turn the data around to get 1000010 or >42 (using) for hex). C (>43 hex) sent right after B might look like:

1110010000110110000111111111
iiiiBBBBBBesCCCCCceiiiiii

Since this is an asynchronous line, the letter C could start at any time after the end of the stop bit for the B. For instance, the two letters could look:

111001000011101100001111111111
iiiiBBBBBBeisCCCCCceiiiiii

Here I have included one idle bit time between the characters. In fact this idle time need not be any even multiple of the bit time — it might be 1/3 of the time, for instance.

Note again that B (binary 100 0010=>42) is backwards (0100 001). To summarize, the no parity transmission formats look like:

Setting	Format	example - 'B'
7/none/1:	sdddddde	001000011
7/none/2:	sdddddeee	0010000111
8/none/1:	s0ddddde	0010000101
8/none/2:	s0dddddeee	00100001011

At this point we should consider what might happen if the sending computer is using a different format than the receiving computer. If the BBS is sending in 7/none/2 the first stop bit will be received as a data bit if you're using 8/none/1. No error indication will appear — the extra stop bit sent serves to ward off the framing error. If you do this on most computers you'll get graphic characters displayed instead of your text because all characters received will be in the range 128 to 255.

So let's try building a table of combinations:

receiver \	7n1	7n2	8n1	8n2	sender
7/none/1	Y	Y	F	F	
7/none/2	T	Y	F	F	
8/none/1	F	G	Y	Y	
8/none/2	F	F	T	Y	

Y=transmission good. Extra sender stop bits look like idles to the receiver.

F=framing error possible. When the sender is sending 8 data bits and the receiver is looking for a stop bit after only 7 data bits a framing error will occur on every byte that doesn't have its top bit set (such as all ASCII characters). When the host is sending 7 data bits and the receiver is looking for 8 an error may or may not occur (depending on the interval between the sending of one character and the next) but the MSB (most significant bit) will be set incorrectly.

G=MSB incorrectly set (see above) always, but no indication of framing error.

T=Normally received correctly. Some possible timing problems. Normally the UART (Universal Asynchronous Receiver/Transmitter) only checks the first stop bit while receiving while honoring both stop bits when transmitting (the 9902 which does this sort of work for the 99/4A does) and this works great. Some equipment (I know of such a device, a digitizer) needs the extra time to deal with the character and will not work right unless both stop bits are sent.

Note that the treatment of stop bits is not symmetric. If the host system needs 2 when IT is getting data you will have to send 2 even though you will be able to read everything the host sends if you use 1 stop bit for receiving.

The next thing we should look at is what can go wrong with the no-parity communications styles discussed so far. There are two types of errors that can get introduced into the bit stream. The first is a loss of carrier. On most modems this will lead to a '0' signal on the line to the computer. So what will this result in? First, suppose that the line is otherwise in the idle state.

In the idle state the line looks like:

```
'1111111111111111' ...
```

a 1-bit error looks like:

```
'111111011111111111' ...
```

When this happens the computer will mistake the error bit for a start bit. It will then grab the next 7 or 8 bits as data and will test the stop bit. This bit will be at the idle state of 1 and therefore will seem okay. So the computer will receive an ASCII 127 or 255 character (depending on whether you are in 7 or 8 data bits). This is the most common error encountered on a bad line. It results in your getting extraneous DEL characters. For exactly this reason most computers do not use DEL chars. Most systems ignore DELs, with the

major exception of DEC machines. When you are on a bad line to a DEC machine you can type 3/4 of your line, and then spurious DELs will delete the chars as fast as you can retype them. You can see why most systems use BACKSPACE (8 ASCII) to delete characters. The DEL character may be printable on your terminal or not — often DELs will print as solid blocks.

If the carrier is lost for 2 bit times the receive will get data bits '0111 1111' for ASCII 126 (remember, once again, that the bits are transmitted in reverse order). Similarly carrier losses for 3, 4 and more bit times can happen. When you get such losses you will see the following characters on your screen:

```
1 2 3 4 5 6 7 8  
DEL ~ ) x o ' @ NUL
```

These are all assuming 7 data bits. I have listed these characters because you will also find these appear if you are using the wrong transmission speed (more often than others anyway).

Bit errors that occur during character transmission cannot be detected when parity is not used, except that a framing error will occur if the stop bit is not 1 as expected.

Aside from loss-of-carrier errors modems are slightly susceptible to noise on the line turning 1 bits into 0 bits and 0 bits into 1 bits. This type of error is somewhat similar to carrier loss. I don't find it to be nearly as much of a problem, as it is rather rare. Often this problem is addressed by using FULL DUPLEX communications, at least to the extent that this helps you check to see whether the characters that you typed were received correctly by the host.

Typical error conditions have been examined. In order to detect some errors parity is often used in data communications. I will continue this article by expanding on the basic information presented so far and introducing the use of parity bits.

Parity for RS232 communications only really comes in two forms: EVEN and ODD. A good terminal program will have five settings you can select for parity, however: EVEN, ODD, NONE, MARK and SPACE. The NONE setting, explained above, means that no bits are inserted after the data bits and before the stop bit(s). The SPACE setting instructs the terminal to send an extra 0 bit between the data bits and the stop bit(s). The MARK setting instructs the terminal to send an extra 1 bit between the data bits and the stop bit(s). Once again, examples are in order. In these examples, the number of data bits is assumed to be 7 and the number of stop bits to be 1. These are typical settings whenever a parity setting other than NONE is used. As well, I have left

the start bit off (it is always a 0 bit to start transmission):

Parity Setting	letter 'B'	letter 'C'
NONE	01000011	11000011
MARK	01000011	11000011
SPACE	01000010	11000010
EVEN	01000010	11000011
ODD	01000011	11000010

The MARK and SPACE settings can be used to simulate 8 data bit transmission when your terminal has no means to change the 7 data bit option. For instance, if you are communicating with a system that expects 8/none/1 you can use 7/space/1 as long as you are only transmitting textual data (i.e. the ASCII characters, 0 to 127). The MARK setting is able to imitate a double stop bit, if you have no control over stop bits.

The EVEN and ODD parity settings generate the extra bit in such a way that the number of 1 bits in the data bits and parity bit adds up to an EVEN or ODD number respectively. For instance, for 'B' the data bits are 0100001. If you are communicating in EVEN parity, the parity bit will be a 0, leading to a total bit count of 2. If you are in ODD parity, a parity bit of 1 will be generated to make the total bit count the odd number 3. As can be seen above, for 'C' the reverse is true. The following shows the parity setting under which the start of the alphabet will give a 0 parity bit:

```
A B C D E F G H I J K
E E E E E E E E E
```

So, if the host is transmitting in EVEN parity and you are receiving in 8/NONE, you will be able to read the letters A, B, D, G, H, K, ... but not C, E, F, I, J. Therefore, if you log on to an unfamiliar system, try using 8/NONE first. If you find that you can read only half the letters, the system is sending using EVEN or ODD parity. By taking any one of the characters that you can read and working out the binary value it is easy to figure out what parity setting is needed. By the way, it is interesting to note that the lower-case letters have exactly the opposite parity of their capital counterparts.

What parity is used most often? Well, if you remember, the most frequent transmission error is a single bit loss of carrier. The most useful parity then is the one that detects that error. For 7 data bits, you will find that ODD parity will detect an error for a single bit loss of carrier, whereas EVEN will not.

Alpha Lock

R.A.Green
Ottawa U.G. Newsletter

Have you ever started playing a game then have to restart because you forgot to release the Alpha Lock? Well, here is a little trick that Assembler Language programmers can use to warn the player, and then wait till s/he releases the Alpha Lock. It will make a programme a little more user friendly.

The following Assembler source programme demonstrates the technique - it literally looks at the Alpha Lock.

Assembler Source

```
*TITLE: ALPHA LOCK DETECTION
*AUTHOR: R.A. GREEN
*
DEF ALPHA Define entry point
REF VMBW VDP Multi Byte Write
*
ALPHA LI R0,0062 "ALPHA LOCK TEST"
LI R1,MSG1
LI R2,15
BLWP @VMBW
TEST CLR R12 CRU Base address of zero
SBZ 21
NOP
TB 7 Test for Alpha Lock
JEQ NOTON Jump if Alpha Lock is off
LI R1,MSGON "ALPHA LOCK IS ON"
MSG LI R0,00A2
LI R2,17
BLWP @VMBW
LIMI 2 Let interrupts in
LIMI 0 So QUIT key will work
JMP TEST Loop
NOTON LI R1,MSGOFF "ALPHA LOCK IS OFF"
JMP MSG
MSG1 TEXT 'ALPHA LOCK TEST'
MSGON TEXT 'ALPHA LOCK IS ON'
MSGOFF TEXT 'ALPHA LOCK IS OFF'
END
```

Corrections and comments by Frank Geitzler

This article has been reprinted from last month's issue. An oversight with the text formatter caused DP-codes to appear as labels, and a greater-than symbol (>) was omitted from two lines. In addition, in two lines the code 'BLWP @VMBW' should be replaced by 'BLWP @VMBW'. It has also been found that if the Alpha Lock is up and the Function key is pressed, this program indicates that the Alpha Lock is down.

DaffyNitions

Terry Atkinson

Well, at this time of year, everyone needs some humor to spice up their lives. So, presented here for your enjoyment, are some quips that I dug up from various sources, including the SOURCE, Computer Haters Handbook, and a couple of my own. Enjoy them, and if you have any of your own, let us know about them.

ABORT: when the bus is full, the conductor shouts "all abort"! (that one may be too technical for you).
ACCURACY: Something impossible for programmers to attain.
ADDER: The part of a cow which counts how much milk is left.
ALPHA: The last name of "Little Rascal" star, Alf.
ANSI: Computer hacker who can't sit still.
ARRAY: Spontaneous exclamation by a hacker when something works for a change. Rarely used.
ASCII: Key to your girlfriends apartment.
AUTO-REPEAT: A key which, when held down, held down, held down....
BARREL PRINTER: A guy who writes XXX on whiskey kegs.
BASIC: Something so simple you need a computer to understand it.
BAUD RATE: Fee charged by loose women; usually \$5 and up.
BIAS: Said of Siamese twins, as in: "They've got a cute little...."
BI-DIRECTIONAL: A computerist who swings both ways.
BIT: Describes computers, as in "OUR" computer cost quite a BIT.
BOOT: What your friends do to you when you brag about your computer.
BUFFER: A nude hacker.
BUFFER AMPLIFIER: One who brags about it.
BUFFER STORE: Where a buffer can't buy anything to wear.
BUG: Small German car found in American Automakers programs in the 50's.
BUG: What your eyes do after staring at a screen too long.
BYTE: Short for BUY IT. Refers to how many peripherals you'll have to purchase to support a computer. e.g. there are 8 "buy it", 16 "buy it" and 64 "buy it" computers.
CHAIN PRINTER: Someone who can't give up printing.
CHIPS: Used to insert into DIP while working at your computer.
COMPUTER SCIENCE: The fastest growing voodoo art course in Colleges.
COPY: What you do at school cause you were playing PARSED so much last night.
CURSOR: What you become when your computer breaks down.

DATA: A nice Italian Girl.
DATA BASE: Where she lives.
DATA BUS: What she drives at work.
DECREMENT: The crap you get from computers.
DEDICATED KEY: What she gave only to you.
DISK: What slips in your back after hours of sitting at a terminal.
DOUBLE DENSITY: Real dumb!
DUMB TERMINAL: A hacker interfaced with a smart terminal.
DUMP: Where all your hobbies go after buying a computer.
DUMP: The best place for computers.
EIGHT BIT CHIP: A one-dollar hooker.
ELECTRONIC MAIL: Post office jargon for anything delivered in less than a week.
ERROR: Made when you walked into the computer store "just to look".
EXECUTION TIME: The time it takes to strangle the salesman who sold you the computer.
EXPANSION UNIT: The room you add to your house to store your computer.
EXPANSION SLOTS: The extra holes in your belt buckle.
FLOPPY: The condition of a user's muscles after sitting around and eating chips.
FLOPPY DISK: Serious curvature of the spine.
FRIENDLY: Said of anything associated with computers that is incomprehensible, or does not work the way it says it will.
GIGABYTE: A painful sting on the Giga.
HACKER: A misanthropic bore.
HARDWARE: Rakes, mowers, and other things you haven't touched this summer.
HEAD CRASH: A collision with a porta-potty.
HEURISTIC: To behave like a heur.
HEXADECIMAL: a 16-letter Shaker curse.
HIGH-LEVEL LANGUAGE: An idiom spoken by hackers wearing tights.
HORIZONTAL SCROLLING: The missionary way.
INCREMENT: What computers eat to produce decrement.
INDUSTRY STANDARDS: Non-conforming guidelines.
INTELLIGENT: A hypothetical term in computing.
INTERLACE: To tie two boots together.
INTERPRETER: The person you take with you to the computer store to understand the salesman. Usually a 12 year old kid.
JOYSTICK: A truncheon used by sadists.
LIGHT PEN: A minimum sentence prison.
LINE FEED: "I've never met anyone as interesting as you before", etc.
MEMORY: A part of a computer where data is placed before destruction.
MULTIPASS: To try again after she turns you down the first time.
MENU: An itemized list of ways to make a mistake on a computer.
MENU: What you'll never see again, cause now you're too poor to eat out.

Secretary Corner

Kevin Fleming

Hello All:

The New Year is upon us already, and we begin another year of meetings and knowledge dissemination. The recent loss of TIBBS and the arrival of Techie, have seemed to balance out somewhat, but we will miss Terry's board.

This meeting in January, marks the start of a new membership year, and the hopeful addition of new members. If you know of anyone who is looking to get more out of their machine, or even if they have just the basic console, get them in touch with us. The larger the group we have, the more of a knowledge base we will have, and the further the knowledge will spread. Keep in mind the new members will need more encouragement to use their machines, and if in the process of expansion, will need information on where to obtain third party hardware and software.

Our main goal is to spread the fellowship of TI users to all those who own, and hopefully wish to know more about their machine. Our group should try to expand it's membership more actively. The new machine from Myarc (we hope this machine will see the light of day), the third party peripherals and software are all available to any who are interested. We have some great support for a really good machine, and our group should more actively pursue memberships.

Keep in mind the start of the new membership year, and renew early.

Your Sec/Treas

Kevin Fleming

455-3320

(Call after 5 but before 9:30 pm)

MNEMONIC: Said of someone suffering from mnemonia.
NYBBLE: What an unsuspecting customer does to a line dangle by a salesman.
OVERSTRIKE: To tempt fate, e.g. Air Traffic Controllers.
OHM: Where the 'heart is!
OUTPUT DEVICE: A word-processor who can't say "No!".
PAPERTAPE PUNCH: A mushy drink that sticks to the roof of your mouth.
PERIPHERAL: Anything that costs a lot of money that can be remotely associated with computers.
PITCH: Adding all the characters in an inch of type will give you the sum of the pitch.
PLOTTER: Computer salesman who soots you browsing during your lunch hour.
POLARITY: Solidarinoscz.
PROGRAM: A random accumulation of bugs.
PROGRAM: What you used to watch on the TV, until you hooked the computer to it.
QWERTY: To be a little strange.
RAM: Where most of the bugs are kept.
RAM: What you do to the side of your computer when it's broken.
ROM: Where you put all the bugs that don't fit into RAM.
READ/WRITE HEAD: Men's room with invitational graffiti wall.
REDUNDANCY: Two computer experts, when none will do.
RETURN: What you do with the computer after RAM doesn't work.
ROUTINE: A program that never works the same way twice.
SKEW: Interface between two consenting computers.
SKEW FAILURE: Premature calculation.
SMART TERMINAL: The one that gets you to buy it.
SOFTWARE: What hackers wear under their hardware.
SUBROUTINE: "Dive! Dive!"
SUPERFICHE: Moby Dick, Jaws, etc.
ULTRAFICHE: Bigger, faster, and harder to land than superfiche.
VARIABLE: Anything with a fixed value.
WINDOW: What you throw the computer thru when you can't RETURN it

Simple "fly over bomb town" game

```
100 CALL CLEAR::CALL HCHAR(23,2,30,30)::CALL COLOR(9,4,4)::CALL HCHAR(24,1,
96,32)::CALL SPRITE(#1,43,2,35,256,0,-10)
110 CALL KEY(0,K,S)::CALL SOUND(-1000,-3,0,200,10)::IF S=0 THEN 110::CALL
POSITION(#1,XX,YY)::AA=XX::CALL SPRITE(#2,46,2,XX,YY)
120 AA=AA+4::CALL LOCATE(#2,AA,YY)::IF AA=21*8 THEN CALL SOUND(100,-7,0)::
CALL POSITION(#2,R,S)::CALL HCHAR(23,S/8+1,32)::AA=0::CALL DELSPRITE(#2)::
GOTO 110
130 CALL SOUND(-100,-3,0,2000-(AA*4),10)::GOTO 120
```